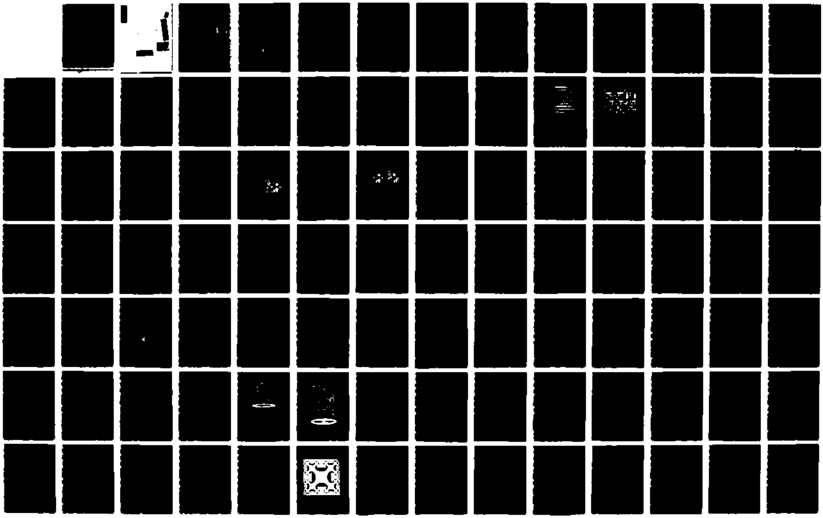
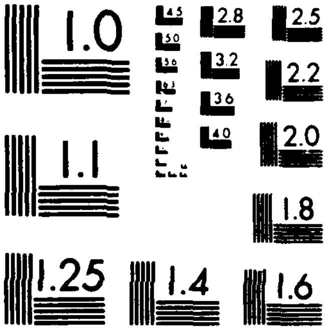
AD-A186 990 SINGLE-LAYER WIRE ROUTING(U) MASSACHUSETTS INST OF TECH 1/4
CAMBRIDGE LAB FOR COMPUTER SCIENCE F M MALEY AUG 87
MIT/LCS/TR-403 N00014-80-C-0622

UNCLASSIFIED F/G 9/1 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE

11

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION
Unclassified

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE

3. DISTRIBUTION / AVAILABILITY OF REPORT
Approved for public release; distribution is unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
MIT/LCS/TR-403

5. MONITORING ORGANIZATION REPORT NUMBER(S)
N00014-80-C-0622

6a. NAME OF PERFORMING ORGANIZATION
MIT Laboratory for Computer Science

6b. OFFICE SYMBOL *(If applicable)*

7a. NAME OF MONITORING ORGANIZATION
Office of Naval Research/Department of Navy

6c. ADDRESS *(City, State, and ZIP Code)*
545 Technology Square
Cambridge, MA 02139

7b. ADDRESS *(City, State, and ZIP Code)*
Information Systems Program
Arlington, VA 22217

8a. NAME OF FUNDING / SPONSORING ORGANIZATION
DARPA/DOD

8b. OFFICE SYMBOL *(If applicable)*

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS *(City, State, and ZIP Code)*
1400 Wilson Blvd.
Arlington, VA 22217

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO

DTIC
ELECTE
NOV 18 1987
S A D

11. TITLE *(Include Security Classification)*
Single-Layer Wire Routing

12. PERSONAL AUTHOR(S)
Maley, F. Miller

13a. TYPE OF REPORT
Technical

13b. TIME COVERED
FROM ________ TO ________

14. DATE OF REPORT *(Year, Month, Day)*
August 1987

15. PAGE COUNT
361

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
channel routing, compaction, computational geometry, constraint solving, covering space, homotopy, global routing, graph algorithms, jog insertion, river routing, routablity,

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

This dissertation concerns the problem of routing wires on a single layer of an integrated circuit or printed circuit board, starting from a sketch of the layer. A sketch specifies the positions of layout features and the topology of the interconnecting wires. Efficient algorithms are presented that (1) determine whether a sketch is routable, and (2) produce for a routable sketch a proper routing that minimizes both individual and total wire length. Both algorithms run in time $O(n^2 \log n)$ on input of size n, and both are simple to implement. They can be adapted to a variety of wiring models, and they subsume most of the polynomial-time algorithms in the literature for single-layer routing and routability testing.

The algorithms are based on two theorems concerning the routings of a sketch. One states that a sketch is routable if and only if for each cut between fixed features, the total amount of wiring forced to cross the cut is no greater than the length of the cut. The second theorem states that every routable sketch has a routing that simultaneously minimizes

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED □ SAME AS RPT □ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL
Judy Little, Publications Coordinator

22b. TELEPHONE *(Include Area Code)*
(617) 253-5894

22c. OFFICE SYMBOL

83 APR edition may be used until exhausted
All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

☆U.S. Government Printing Office: 1985—507-047

Unclassified

18. routing, topology, VLSI layout, wiring, wire length minimization.

19. the length of every wire, and it characterizes the wires in this routing. To formalize and prove these theorems, a rich mathematical theory of single-layer wire routing is developed. Its central tool, which is new to the wire-routing literature, is the lifting of wires and cuts to a simply connected topological covering space of the routing region.

As another application of this theory, the thesis presents a general algorithm for one-dimensional layout compaction. Given a routable sketch, it finds a proper sketch of minimal width obtainable by displacing the features horizontally and moving the wires, always maintaining routability. Thus it automatically inserts into wires all jog points that help in compressing the layout. In the worst case the compaction algorithm uses time $O(n^4)$ and space $O(n^3)$ on input of size n. The technique on which algorithm is founded is nearly independent of the wiring model, and it applies to many-layer as well as single-layer compaction problems.

Accession For
NTIS GRA&I ☑
DTIC TAB □
Unannounced □
Justification
By
Distribution/
Availability Codes
Dist | Avail and/or Special
A-1

DTIC COPY INSPECTED 6

Single-Layer Wire Routing

F. Miller Maley
MIT Laboratory for Computer Science

Submitted on 7 August 1987 to the Department of
Electrical Engineering and Computer Science
at the Massachusetts Institute of Technology
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

© F. Miller Maley, 1987

The author hereby grants to MIT permission to reproduce and to distribute copies of this thesis document in whole or in part.

Author's address as of September 1987: Department of Computer Science, Princeton University, Princeton. NJ 08544.

This research was supported in part by a Graduate Fellowship from the Office of Naval Research, and in part by the Defense Advanced Research Projects Agency under contract N00014-80-C-0622.

7 11 17 113

Single-Layer Wire Routing

by

F. Miller Maley

Submitted on 7 August 1987 to the Department of
Electrical Engineering and Computer Science
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

Abstract

This dissertation concerns the problem of routing wires on a single layer of an integrated circuit or printed circuit board, starting from a *sketch* of the layer. A sketch specifies the positions of layout features and the topology of the interconnecting wires. Efficient algorithms are presented that (1) determine whether a sketch is routable, and (2) produce for a routable sketch a proper routing that minimizes both individual and total wire length. Both algorithms run in time $\Theta(n^2 \log n)$ on input of size n, and both are simple to implement. They can be adapted to a variety of wiring models, and they subsume most of the polynomial-time algorithms in the literature for single-layer routing and routability testing.

The algorithms are based on two theorems concerning the routings of a sketch. One states that a sketch is routable if and only if for each *cut* between fixed features, the total amount of wiring forced to cross the cut is no greater than the length of the cut. The second theorem states that every routable sketch has a routing that simultaneously minimizes the length of every wire, and it characterizes the wires in this routing. To formalize and prove these theorems, a rich mathematical theory of single-layer wire routing is developed. Its central tool, which is new to the wire-routing literature, is the lifting of wires and cuts to a simply connected topological covering space of the routing region.

As another application of this theory, the thesis presents a general algorithm for one-dimensional layout compaction. Given a routable sketch, it finds a proper sketch of minimal width obtainable by displacing the features horizontally and moving the wires, always maintaining routability. Thus it automatically inserts into wires all jog points that help in compressing the layout. In the worst case the compaction algorithm uses time $O(n^4)$ and space $O(n^3)$ on input of size n. The technique on which the algorithm is founded is nearly independent of the wiring model, and it applies to many-layer as well as single-layer compaction problems.

Key words: channel routing, compaction, computational geometry, constraint solving, covering space, homotopy, global routing, graph algorithms, jog insertion, river routing, routability, routing, topology, VLSI layout, wiring, wire length minimization.

Thesis Supervisor: Charles E. Leiserson
Title: Associate Professor of Computer Science and Engineering

Preface

This dissertation is the product of a four-year study on the general problem of wire routing under separation and homotopy constraints. Originally intended as a master's thesis, the project quickly grew out of control when repeated attempts to solve the fundamental problems ended in failure. The driving force behind the growth was a desire for mathematical rigor. I devised the central algorithm of this thesis, the sketch routing algorithm, and was convinced of its correctness, long before finding any technical justification for it. All attempted correctness proofs using elementary tools broke down, and the breakdowns could be traced to a single source: a lack of technical tools for dealing with the concept of homotopy at the heart of the routing problem. Since homotopy is a topological notion, I turned to algebraic topology, and thus was born the theory that accounts for the bulk of this thesis.

Though my approach to single-layer wire routing has been lengthier and more involved than one might like, I expect it to support further fruitful work on wire routing. both practical and theoretical. This research has had two goals: to establish certain theorems and algorithms concerning wire routing and compaction, and to blaze a trail through the vast terrain between homotopy theory and circuit design. The tension between these aims accounts for the technical depth of this study. To read it carefully is likely to be a laborious task; yet I hope scholars of algorithms will find it rewarding. Being a worked-out example of wire routing in two specific models. this dissertation may serve as a source of ideas and a prototype for studies of other models of wiring. Subsequent treatments should be simpler, or at least easier, with the steps and missteps of this thesis as a guide. And as a first cut at a theory of single-layer routing, it demonstrates the power of bringing topological concepts to bear on routing problems.

Organization and prerequisites

Because this thesis addresses topics that run from topology through algorithms and circuit design. I have tried to make it accessible to specialists and students in several areas. The danger, of course, is that I might make it accessible to nobody. To guard against that possibility, I have separated the algorithms from the underlying mathematics, and confined the advanced topology to a pair of chapters, namely Chapters 2 and 3. The glossary includes definitions of mathematical terms that may be unfamiliar, and I have provided a table of notations on pages 10–12.

Those who are primarily interested in wire routing and compaction should read Chapter 1, which shows how to solve routability and routing problems, and Chapter 9, which presents and justifies a compaction procedure. Chapter 10 discusses refinements and extensions of these algorithms. When describing algorithms, I assume some knowledge of the techniques of computational geometry and algorithmic graph theory.

Those who are interested in the application of topology to routing problems should read the remaining chapters, beginning with the definitions in Chapter 2. The core of this dissertation is the development of a theory of single-layer wire routing in Chapters 3 through 7. Chapter 8 uses this theory to derive results about the sketch model. Most of these chapters require familiarity with point-set topology; a knowledge of elementary homotopy theory is also helpful. For those with no prior exposure to algebraic topology, I have provided a short introduction to homotopy theory in Chapter 2.

Acknowledgements

This thesis, and the degree it represents, could never have been completed without the help and support of three people. One is my mother, Ann Maley, whose love and encouragement kept me going at the most difficult times. Another is my thesis advisor, Charles Leiserson, who, along with his former student Ron Pinter, provided the starting point and the motivation for this entire line of research; they discovered the connection between routability conditions and compaction with automatic jog insertion. Chapter 1 is derived from a joint paper [21] with Prof. Leiserson. Charles also taught me how to write and speak on technical matters, and has contributed greatly to this exposition in innumerable discussions. The third is John Baez, my former roommate and good friend, who rekindled my interest in pure mathematics by introducing me to algebraic topology. Discussions with John convinced me to take seriously my idea of using covering spaces to study wire routing. Without the inspiration that arose from those and further conversations, my program of research would have stagnated. I wish to express my heartfelt appreciation for all that these people have done for me.

Many other people have contributed to this thesis in great and small ways. I especially thank my other two readers, Prof. David Anick and Prof. Bill Dally, for their willingness to serve on my thesis committee, and their helpful comments

and referrals. In a pair of wonderful courses (er, subjects), Prof. Anick taught me everything I know about algebraic topology, and for this too I am grateful. Thanks also to Bonnie Berger, Bard Bloom, Ray Hirschfeld, Joe Kilian, Bruce Maggs, and Su-Ming Wu for reading and pasting figures into various drafts of the thesis. Their help was instrumental in finishing the final copy.

Part of the inspiration for this research came from the success of actual programs. Encouraging experimental results were provided by Michel Doreau of Digital Equipment Corporation, whose use of cut constraints in his PCB router "TWIGGY" first suggested the sketch routability theorem. Some special-case compaction algorithms developed by Leiserson and Pinter were implemented by Andrew Hume of at AT&T Bell Laboratories, and used for channel routing.

Over the years, several people have offered technical advice on improving my proofs and presentations, among them John Baez, Ravi Boppana, Ron Greenberg, and Johan Hastad. I am grateful for their insights. I would also like to thank the referees and editors of my published papers, particularly Prof. Thomas Lengauer, for their comments and encouragement.

Readers will note that I take pleasure in inventing fanciful and amusing terminology. I cannot take all the credit for my strained analogies and mixed metaphors, however; those who suggested terms I later adopted include Ray Hirschfeld (who is responsible for 'blanket'), Phil Klein, Charles Leiserson, Mark Newman ('embedding' into a sheet), and Alan Sherman.

Thanks to Bard Bloom and Joe Kilian for providing places to live when I overran the Spring 1987 thesis deadline ...and to Albert Meyer for his comfortable couch.

Finally, I wish to acknowledge the organizations and people who supported my research environment. This thesis was carried out using office space and computing resources of the Theory of Computation Group of the MIT Laboratory for Computer Science. I thank all the members of the TOC group for making it a great place to work. I am especially grateful to its underrecognized software and hardware gurus, Ray Hirschfeld and Mark Reinhold, for their constant support of our computing environment and their willingness to answer any computer-related question, however naive. Thanks also to the Academic Computer Center at Amherst College and its friendly denizens for computer time and blackboard space. Funding for research-related expenses was provided by the Defense Advanced Research Projects Agency. Last but far from least, I thank the Office of Naval Research and the American Society for Engineering Education for the generous fellowship that has supported me over the past four years.

F. M. M.
Cambridge, Massachusetts
August, 1987

Table of Contents

Introduction: **Single-Layer Wire Routing** **13**

A. Background 14

B. Thesis Overview 20

Chapter 1: **Sketch Algorithms** **28**

1A. The Sketch Model 28

1B. The Rubber-Band Equivalent of a Sketch 33

1C. Testing the Routability of a Sketch 40

1D. Routing a Sketch 44

1E. Efficiency Concerns 50

1F. Faster Routability Testing 54

Chapter 2: ***Topological Preliminaries*** **62**

2A. Homotopies and the Fundamental Group 65

2B. Covering Spaces 69

2C. Paths and Loops in the Plane 76

2D. Topological Manifolds 79

Chapter 3: **The Topology of Blankets** **84**

3A. Constructing Paths in Blankets 88

3B. Separation Results 93

3C. Properties of Separations 99

3D. Elastic Chains in Sheets 104

Chapter 4: **Flow Across Cuts and Half-Cuts** **110**

4A. The Design Model 111

4B. Flow: A Characterization of Congestion 115

4C. Relations Among Cuts and Wires 122

4D. Properties of Flow 128

4E. The Branches of a Blanket 135

4F. Safety of Cuts and Half-Cuts 140

Chapter 5: **Routing a Safe Design** **148**

5A. Construction of Ideal Routes 152

5B. Ideal Routes Are Taut 160
5C. Ideal Routes Form a Design 168
5D. Ideal Designs Are Properly Connected 171
5E. Ideal Wires Are Self-Avoiding 176
Chapter 6: **Routability Conditions for Designs 182**
6A. Unsafe Designs Are Unroutable 183
6B. Ideal Embeddings Have Optimal Length 189
6C. Summary of Design Theorems 194
6D. Cuts That Decide Routability 196
Chapter 7: **From Theory to Algorithms 203**
7A. Geometric Representations of Path Classes 204
7B. Crossing Sequences 211
7C. Two Methods for Computing Plans 218
7D. The Geometry of Ideal Wires 226
7E. Routing Through a Maze 235
Chapter 8: **Return to the Sketch Model 242**
8A. The Correspondence Between Designs and Sketches 243
8B. Sketch Theorems 249
8C. Correctness of the Sketch Algorithms 255
Chapter 9: **Sketch Compaction 261**
9A. Problem Statement 265
9B. Computing Flows During Compaction 268
9C. The Compaction Algorithm 275
9D. The Adjacency Graph of a Sketch 281
9E. The Abstract Compaction Algorithm 287
9F. Implementing the Abstract Algorithm 294
Chapter 10: **Extensions of the Theorems and Algorithms 300**
10A. Representation Issues 303
10B. Wiring Rules and Wiring Norms 310
10C. The Terminals of Traces 318
10D. An Alternative to the Sketch Model 325
Conclusion: **A Critical Review 331**
A. Summary of Results 331
B. Directions for Future Research 335
Glossary 339
References 358

Table of Symbols

For the most part, uppercase Roman letters denote data structures or topological spaces, lowercase Roman letters represent points in those spaces, lowercase Greek letters denote paths, and uppercase Greek letters represent sets of paths or points.

Symbol	Meaning	Page
$\overline{xy}$	Line segment with endpoints x and y	17
$\\|\cdot\\|$	A norm, the *wiring norm*, on the plane R^2	29
$\\|P-Q\\|$	Distance between the regions P and Q	30
$\vert D\vert$	Size of the data structure D	34
$O(f)$	Functions g such that $\vert g(n)\vert \leq cf(n)$ for some c as $n \to \infty$	34
$\Omega(f)$	Functions g such that $g(n) \geq cf(n)$ for some $c > 0$ as $n \to \infty$	44
$\Theta(f)$	Functions in both $O(f)$ and $\Omega(f)$	50
I	Unit interval $[0,1]$	62
R^1	Real line	62
R^2	Cartesian plane	62
H^2	Upper closed half-plane of R^2	63
S^1	Unit circle in the plane R^2	63
$Int\ A$	Topological interior of A	63
$Cl\ A$	Topological closure of A	63
$Fr\ A$	Frontier or topological boundary of A	63
$t \mapsto E(t)$	Function that takes t to $E(t)$	63
$F(x,\cdot)$	Function that takes y to $F(x,y)$	63
$f\vert_U$	Restriction of f to U	63
id_X	Identity map on the space X	63
$Im\ \phi$	Image of the function ϕ	63
$\alpha : A \rightsquigarrow B$	The path α runs from A to B	63
$Mid\ \alpha$	Middle $\alpha((0,1))$ of the path α	63
$x \triangleright y$	Linear path from x to y	63
$\alpha_{a:b}$	Subpath of α from $\alpha(a)$ to $\alpha(b)$	64
$\alpha \star \beta$	Concatenation of α and β	64

Symbol	Meaning	Page
$\widehat{\alpha}$	Reverse of the path α, namely $\alpha_{1:0}$	64
$\Vert\alpha\Vert$	Arc length of α in the norm $\Vert\cdot\Vert$	64
$\lvert\alpha\rvert$	Euclidean arc length of α	64
$\alpha \simeq_P \beta$	The paths α and β are path-homotopic	65
$[\alpha]_P$	Set of paths that are path-homotopic to α	66
$[\alpha]_P \star [\beta]_P$	Equivalent to $[\alpha \star \beta]_P$	66
$\pi_1(X, x_0)$	Fundamental group of X at x_0	66
$f_\star$	Homomorphism of fundamental groups induced by f	67
$Ker\ \phi$	Kernel of the homomorphism ϕ	68
$f \simeq g$	The maps f and g are homotopic	68
$\widetilde{g}$	Lift of the map g	71
$inside(\lambda)$	Inside of the simple loop λ	76
$outside(\lambda)$	Outside of the simple loop λ	76
$Bd\ M$	Boundary of the manifold M	79
$Bd\ f$	Restriction of f to the boundary of its domain	80
$\bigoplus P_i$	Topological sum or disjoint union of the spaces P_i	80
$\alpha \simeq_L \beta$	The links α and β are link-homotopic	88
$[\alpha]_L$	Set of links homotopic to α	88
$left(\alpha)$	Left scrap of the simple link α	99
$right(\alpha)$	Right scrap of the simple link α	99
$width(X, \Omega)$	Width of the detail X in the design Ω	113
$cap(\chi, \Omega)$	Capacity of χ in the design Ω	114
$cross(\alpha, \beta)$	Number of crossings between α and β	114
$tangle(\chi, \omega)$	Entanglement of the link ω with the cut χ	114
$cong(\chi, \Omega)$	Congestion of the cut χ in the design Ω	114
$wind(\chi, \omega)$	Winding of the links χ and ω	118
$flow(\chi, \Omega)$	Flow across the link χ in the design Ω	119
$margin(\chi, \Omega)$	The difference $cap(\chi, \Omega) - flow(\chi, \Omega)$	140
$P(\sigma)$	Polygon $\{\, x : \Vert x - \sigma(0)\Vert = \Vert\sigma\Vert \,\}$	226
$\dot{\alpha}$	Angle at which the path α travels	227
$\delta^\perp$	Angle of the segment of the unit polygon ending at δ	227
$\delta^\top$	Angle of the segment of the unit polygon starting at δ	227
$[\delta, \theta]$	Angles lying between δ and θ clockwise	228
$R(\dot{\sigma})$	Interval $[\dot{\sigma}^\perp, \dot{\sigma}^\top]$	228
$L(\dot{\sigma})$	Interval $[-\dot{\sigma}^\perp, -\dot{\sigma}^\top] = R(-\dot{\sigma})$	228
$\flat_\epsilon$	Transformation from sketch model to design model	244
$\sharp_\epsilon$	Transformation from design model to sketch model	244
$\theta^\flat$	Abbreviation for $\flat_\epsilon(\theta)$	244

Symbol	Meaning	Page
$\omega^\sharp$	Abbreviation for $\sharp_\epsilon(\theta)$	244
$\alpha^\natural$	Abbreviation for $\sharp_\epsilon(\flat_\epsilon(\theta))$	244
$\mu(p)$	Number of the module that contains the point p	265
x_p, y_p	Coordinates of the point p	265
$p(\mathbf{d})$	Point to which configuration $\mathbf{d}$ moves the point p	265
$\Delta_{pq}(\mathbf{d})$	The x-coordinate of $q(\mathbf{d})$ minus that of $p(\mathbf{d})$	265
$\Delta_{PQ}(\mathbf{d})$	Difference in displacements of the modules $\mu(Q)$ and $\mu(P)$	266
$\mathbf{C}(S)$	Configuration space of the modular sketch S	266
$h \diamond S$	Image of sketch S under the homeomorphism $h\colon R^2 \rightarrow R^2$	266
χ_{pQ}	Critical potential cut from feature endpoint p to feature Q	275
ϕ_{pq}	Potential cut $\mathbf{d} \mapsto p(\mathbf{d}) \triangleright q(\mathbf{d})$	276

Introduction

Single-Layer Wire Routing

A problem that frequently arises in the design of computer components is that of routing wires through some interconnection medium. Most wire-routing problems are computationally hard: determining whether an instance of a routing problem is even solvable is usually NP-complete. In this thesis I show that if the wires are restricted to a single planar layer, and if rough routings of the wires with respect to the routing obstacles are given, then the wires can be routed efficiently and optimally. 'Efficiently' means that the routing algorithm runs in polynomial time, and 'optimally' means that it simultaneously minimizes the length of every wire. To say it another way: Given the topology of a circuit layer, one can quickly produce a legal and nonwasteful geometry for that layer, or determine that no legal geometry is compatible with the given topology. Figure 1 illustrates this kind of routing problem.

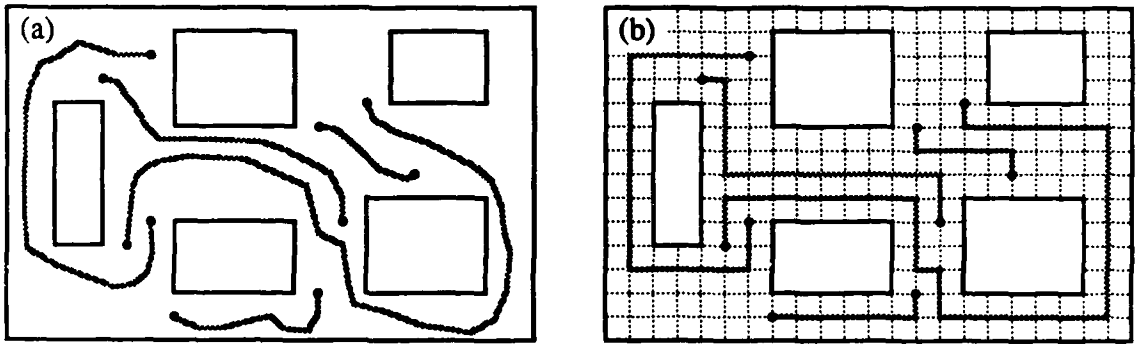


Figure 1. *An instance of a one-layer routing problem.* The wires (grey paths) in the layout of panel (a) are rough routings. They are to be deformed into nonintersecting paths in the grid (dotted lines) shown in part (b), with their endpoints kept fixed and without moving them onto or across any features (dark points and lines). Panel (b) shows a solution with minimum wire length.

The fundamental fact about single-layer wire routing, which I prove, is that local routability conditions are necessary and sufficient for global routability. Consider the layout in Figure 2. The wires cannot be routed: the topology forces too many wires to pass between the obstacles A and B. In other words, the channel between A and B has greater congestion than capacity. The routability of layouts like those in Figures 1 and 2 is completely determined by *the congestions and capacities* of channels. This result leads to efficient algorithms for testing routability, and also to novel algorithms for layout compaction. I present these algorithms here.

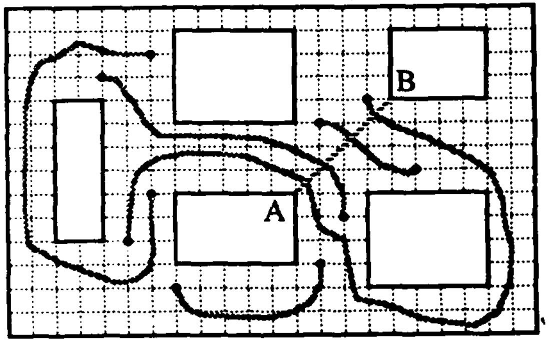


Figure 2. *An unroutable instance.* This layout cannot be routed in the given topology because the channel between obstacles A and B is overfull. More precisely, the *cut* (striped segment) has space for only three crossings by wires, and all four crossings of this cut are necessary, despite the fact that only three distinct wires cross it.

What makes single-layer routing difficult and interesting is the possibility that different parts of the same wire may interact. As Figure 2 shows, a wire can pass *through a channel more than once*, and *the different parts of the wire in that channel* are constrained differently. Thus a wire behaves in some ways like several wires and in some ways like a single wire. To confront this issue I bring in ideas from homotopy theory and show how to analyze single-layer wiring by *lifting* wires and cuts from the routing region to its simply connected covering space. The covering space lets us formalize and work with the notions of the amount of wiring "forced" to pass across a cut, the regions that are "forbidden" to a wire, and the "necessary" crossings of a cut by wires, all of which play major roles in one-layer routing problems.

A. Background

This section puts my routing problem—which will be defined formally in Section 1A—into the context of other wire-routing problems, and it explains how that problem grew out of earlier work. It shows how single-layer routing with rough routings given generalizes the "river routing" problems previously studied, and how further generalizations lead to NP-complete problems. Considerations like these provide the theoretical impetus for my work. The following section offers an outline of the thesis itself and an introduction to its main ideas.

Types of wire-routing problems

Wire-routing problems abound, but they share some common characteristics. The wires, when routed, must connect certain points called **terminals** in a specified pattern, and they must satisfy some geometric constraints such as having a certain minimum thickness and separation from one another. Additional constraints may be imposed on the wires, e.g., that they be composed of rectilinear segments. The space in which wires are to be placed is called the **routing region**. In almost all practical problems, the routing region consists of one or more planes, or **layers**, with wires being allowed to pass between layers only at certain points.

The character of a wire-routing problem depends largely upon the topology of the routing region. Multilayer routing problems are usually NP-complete [51], even when the routing region has a simple shape. For this reason, much of the theoretical work on multilayer wire routing has concentrated on approximation algorithms [1, 4, 45]. These algorithms do not attempt to route within a fixed region, but instead they produce wirings that approach optimality in terms of the routing space or the number of layers they use. Single-layer routing problems are also NP-complete in the general case [20, 44]. Several restricted single-layer routing problems are known to be efficiently solvable, however, including those in which the routing region is simply connected [8, 22, 41, 49, 52] or annular [2] and the terminals lie on its boundary. One can also efficiently route edge-disjoint paths through a planar graph, provided that the terminals lie on a single face of the graph [3, 17, 32, 42]. Such routings are said to be in "knock-knee" mode. One can then convert the edge-disjoint paths into multilayer routings [5, 42].

The tractable routing problems are of three kinds. In a pure routing problem, the routing region and the terminals are fixed; the algorithm must determine whether the wires can be routed, and if so, find feasible **realizations** (or **detailed routings**) for them. In most single-layer routing problems, one can also minimize the length of every wire, which is desirable from a practical standpoint. Sometimes one asks only whether the wires can be routed at all; then one is concerned with a **routability** problem. The NP-completeness results mentioned above apply to routability problems. In a **placement** problem, one thinks of the terminals as being attached to **modules** which can move. As modules move, the shape of the routing region may change. The issue is to find placements for the modules and feasible realizations for the wires so as to minimize some geometric quantity like the area of the routing region.

Wiring models

When studying algorithms for wire-routing problems, one must work at a more abstract level than that of physical devices. One needs a mathematical **wiring model** for the wires and the rules they must obey. For example, wires are usually represented as paths without thickness, but the minimum spacing between the (abstract) wires is increased to allow for the thickness of the actual (physical) wires. If one works with wires of differing thicknesses or materials, then the minimum separation between two wires will depend on which wires they are.

The wiring model most popular among theorists is what I call the **grid-based** model. It achieves simplicity and convenience without hiding any of the essential difficulties of placement and routing. In this model the routing region is overlaid with a rectilinear grid, and wires are required to be disjoint paths within the grid. The spacing between the gridlines corresponds to the minimum separation between wires. Other common models dispense with the grid and allow wires to contain diagonal segments or even circular arcs. Some models also permit different wires to have different separation requirements. My routing problems provide these options, but the examples in this Introduction stick to the grid model.

River routing

A single-layer routing problem that is well understood is the one-layer **river routing** problem [8] as refined by Leiserson and Pinter [22]. I state it for the grid-based wiring model, although other wiring models may be substituted [48]. The routing region is a rectangular **channel**, and the problem is to connect terminals $A_1, \ldots, A_n$ on its bottom edge with corresponding terminals $B_1, \ldots, B_n$ on its top edge. See Figure 3. Wires must be vertex-disjoint paths in the rectilinear grid with integer gridpoints; all the terminals are assumed to have integral coordinates. For technical convenience, the wires are allowed to run along the bottom gridline of the channel, but not along the top. The terminals $B_1, \ldots, B_n$ must be in the same order as $A_1, \ldots, A_n$, or else the wires would have to intersect, since the grid is planar.

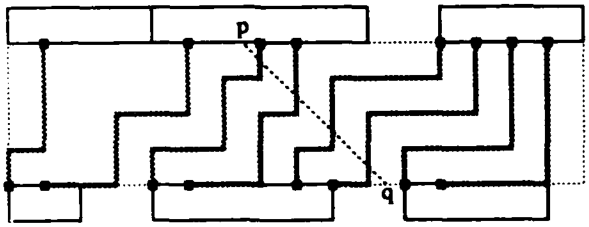


Figure 3. *River routing.* An instance of the problem of river routing in a rectangular channel: connect pairs of terminals (dark points) by nonintersecting wires in a grid (not shown). The grey lines show one feasible set of realizations for the wires. Dotted lines enclose the routing channel. The line from p to q is a *cut* of the channel; it has *congestion* 4 and *capacity* 4.

When the wires can be legally routed, a simple, "greedy" algorithm suffices to find their minimum-length feasible realizations in time proportional to the size of the output. But to determine whether these realizations exist is even easier; one can test routability in time proportional to n, the number of wires.

An instance of the river routing problem can be solved if and only if it satisfies certain easily checked **routability conditions**. Consider a line segment, or **cut**, $\overline{pq}$ that runs from the bottom edge of the channel to the top. Suppose that the terminals to the left of $\overline{pq}$ are A_1 through A_i and B_1 through B_j. Then $|j - i|$ different wires have terminals on opposite sides of $\overline{pq}$, and hence *must cross* $\overline{pq}$. I call the quantity $|j - i|$ the **congestion** of $\overline{pq}$. On the other hand, the number of wires that *can cross* $\overline{pq}$ without touching is equal to the horizontal or vertical separation between p and q, whichever is larger. I call this quantity the **capacity** of $\overline{pq}$ and say that the cut $\overline{pq}$ is **unsafe** [6] if its congestion exceeds its capacity. If any cut in the channel is unsafe, then there is no legal way to route all the wires. Less obvious is the converse: if *no* cut in the channel is unsafe, then there *is* a legal way to route all the wires. In fact, as shown in [22], the wires can be routed unless one of $2n$ special cuts is unsafe. Thus to test routability, it suffices to check $2n$ inequalities of the form

$$\text{congestion of } \overline{pq} \leq \text{capacity of } \overline{pq}.$$

Because the conditions for routability are so simple, one can efficiently solve various placement problems associated with river routing. For example, one can determine how close together the two rows of terminals may be placed while permitting the wires to be routed [8]. If the top row of terminals is free to move relative to the bottom row, then one can find the offset between the two rows that allows the minimum separation between them [34]. Finally, suppose that the terminals on each side of the channel are partitioned into contiguous modules, as in Figure 3, and that each module is free to move horizontally. Then one can position the modules and route the wires so as to minimize the width of the channel [22].

Rough routings

The tractable single-layer routing problems share the property that **rough routings** of the wires can be determined in advance. To have a rough routing ρ of a wire ω means that every feasible realization of ω can be continuously deformed into ρ within the routing region. In mathematical language, every realization of ω is *path-homotopic* to ρ. When the routing region is simply connected, any two paths between the terminals of ω are path-homotopic, and hence any such path serves as a rough routing for ω. When the routing region is ring-shaped, rough routings cannot be chosen arbitrarily, but only a few sets of rough routings need consideration. A routing algorithm can simply try each set, and in fact the algorithm of [2] does just that. In contrast, when the routing region has an arbitrary number of holes, as effectively happens in the NP-complete single-layer routing problems, the number of sets of rough routings that need consideration seems to be exponential.

One is naturally led to consider single-layer routing situations in which rough routings of wires are given. Pinter [41] proposed such a problem, called DRH ("Detailed Routing given a Homotopy"),* which involved routing wires in a finite rectilinear grid. An instance of DRH comprises (1) rectangular modules within a bounding box, (2) terminals on the modules' boundaries, and (3) nonintersecting rough routings that connect pairs of terminals. DRH is a routability problem: it asks whether the given rough routings can be continuously deformed, with their endpoints fixed and without touching any other modules, so that the resulting wires are disjoint paths in the grid.

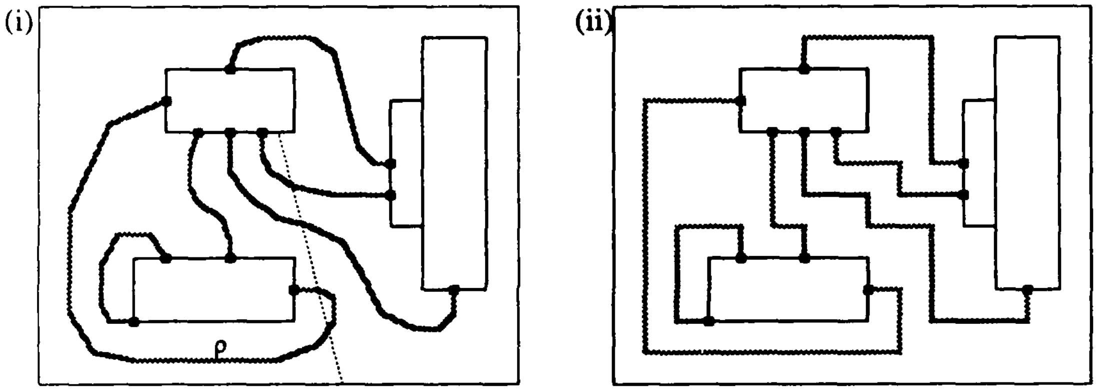


Figure 4. *The problem called DRH.* **Part (i) shows an instance of Pinter's problem DRH. Solid rectangles are modules, dark points are terminals, and grey curves are rough routings. This instance is routable, because the wires can be realized as shown in (ii). (The grid is not shown.) The dotted line in (i) is a cut. Although the rough routings cross it four times, two of those crossings are not necessary, as they can be removed by deforming the rough routing ρ. Hence the congestion of the cut is 2.**

Like the river routing problem, DRH can be analyzed in terms of the congestions and capacities of cuts. One defines a cut to be a line segment whose endpoints lie on modules and whose interior falls in the routing region. The capacity of a cut is the number of wires that can cross the cut without touching; it depends only on the number of gridlines the cut crosses. The congestion of a cut is, in essence, the number of times that wires are forced to cross the cut; it depends upon the topology

[*] By 'homotopy' he meant a set of rough routings, one for each wire to be routed. Technically, the term 'homotopy' refers to a continuous deformation of topological maps.

of the rough routings. As before, we say that a cut is unsafe if its congestion exceeds its capacity. Cole and Siegel [6] showed that an instance of DRH is unroutable if and only if it has an unsafe cut.

The characterization of routability has many applications. It was used in [6] to develop a fast algorithm for solving DRH, given a nethod of computing congestions of cuts. Leiserson and I presented such a method in a subsequent paper [21], thus showing that DRH is solvable in polynomial time. We also set forth routability conditions for a problem very similar to DRH and used them to construct a simplified routability testing algorithm. As in the case of river routing, the routability conditions can be used to solve placement problems as well [29].

Role of this thesis

The fact that DRH is tractable suggests that single-layer *routing* problems may also be efficiently solvable when rough routings are specified. In fact, Leiserson and I proposed a polynomial-time algorithm for such a problem in our paper [21]. Proving the correctness of our algorithm, however, turned out to be much more difficult than we expected. The problem was fundamental: we had almost no technical tools for working with wires in multiply connected regions. The results in [6] concerning DRH worked only for the grid-based wiring model, and even so, their mathematical foundations were unclear. The additional complexity that arises in continuous wiring models is considerable, as one can see by comparing the papers [52] and [22]. All told, the problem of converting rough routings to detailed routings was poorly understood.

In this dissertation I remedy that situation and show how single-layer routing problems can be efficiently solved. My main technical contribution is a mathematically rigorous theory of single-layer wiring. It gives necessary and sufficient routability conditions for DRH-like problems, and it applies to a variety of common wiring models, gridless models included. In addition, it characterizes the minimum-length feasible realizations of wires, thus shedding light on routing problems as well as routability problems. This theory allows me to justify and generalize the routability testing and routing algorithms given in my earlier paper [21].

The theory of single-layer wire routing has applications to placement problems as well. One placement problem of great practical importance is *layout compaction with automatic jog insertion*. This problem generalizes the problem of placing modules for river routing in a channel [22], and it can be solved similarly by means of routability conditions. In my master's thesis [29] I presented a polynomial-time algorithm for this problem, but it was restricted to the grid-based wiring model. Using the new theory of single-layer wiring, I extend this algorithm to many other wiring models.

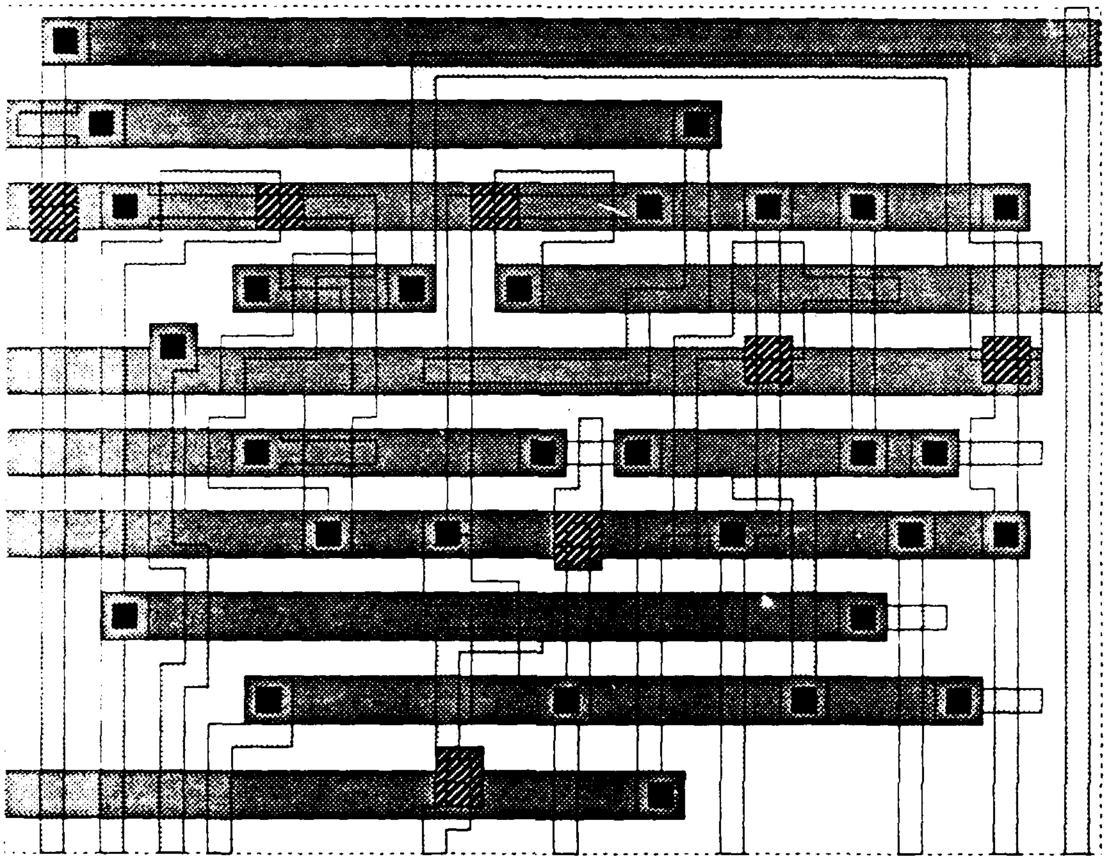


Figure 5. *An integrated circuit layout.* This figure depicts a low-level representation of a portion of an integrated circuit. The layout comprises several layers; each layer is nothing more than a set of polygonal regions. The regions are shaded according to layer, and the shading of upper layers occludes that of lower layers.

B. Thesis Overview

This section outlines the structure of the thesis and describes the main ideas behind each chapter. It also provides some practical motivation for this research beyond the theoretical reasons just discussed. Because the problems I study are not easily defined, a precise statement of my main results must wait until Section 1A.

This thesis studies three problems of single-layer wire routing that arise when rough routings of wires are given. They concern an abstraction of a circuit layer called a *sketch*. The problems are *sketch routability*, *sketch routing*, and a placement problem: *(one-dimensional) sketch compaction*. I present polynomial-time algorithms for all three. These problems seem nearly as general as they can be and still remain efficiently solvable. On the one hand, they subsume most of the single-

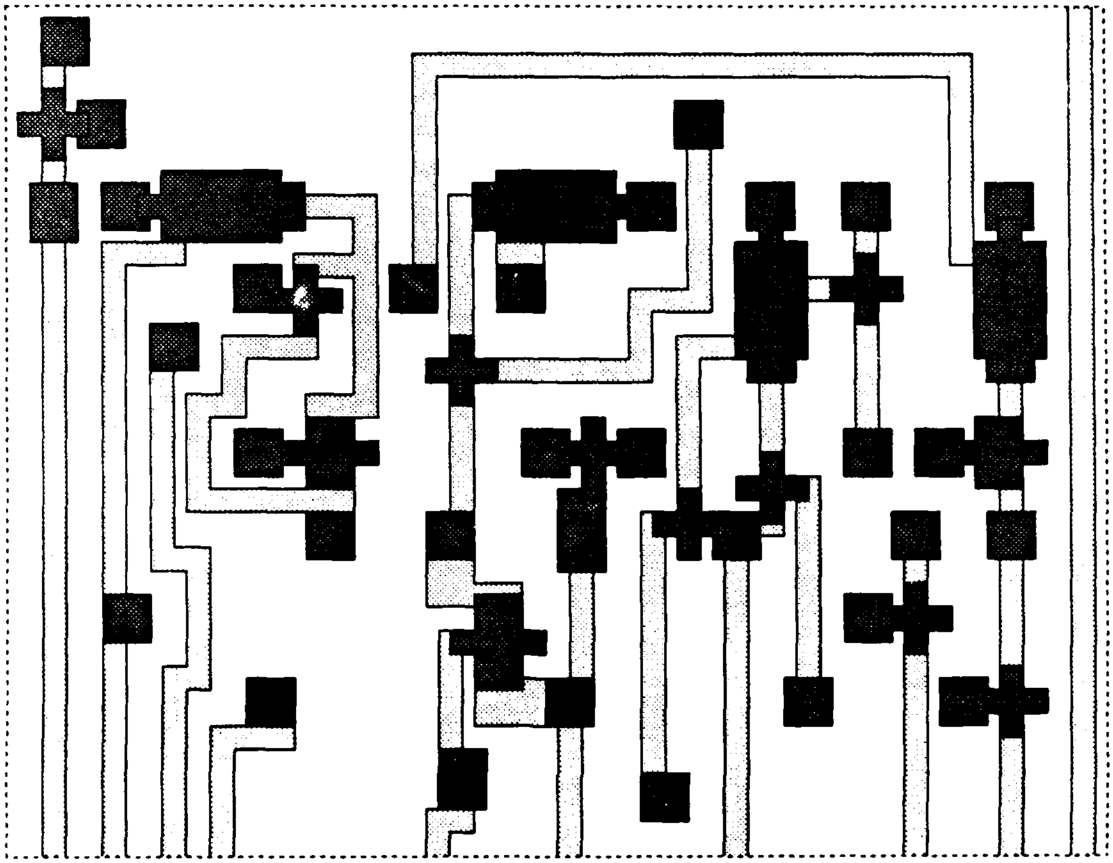


Figure 6. *Rigid and flexible components.* Here is a more abstract representation of the lower layer of the circuit in Figure 5. It distinguishes between rigid devices (dark grey) and flexible wires (light grey).

layer routing and placement problems that have previously been proven tractable. On the other hand, natural variations on these problems that are less restrictive are also NP-complete [41, 48], and hence are unlikely to have polynomial-time algorithms.

Wires as flexible objects

One motivation for the sketch problems stems from the design of integrated circuits (ICs) and printed circuit boards (PCBs). Figure 5 depicts part of the layout for an integrated circuit: each grey tone corresponds to one layer of the chip. This layout contains no explicit information about the functions performed by different regions on a layer. The designer, on the other hand, considers some areas to be wires and other areas to be device components, as shown in Figure 6. He or she is often willing to let wires change in shape and length, but wants to control the shapes of active devices and the widths of wires, as these parameters have the greatest effect on the performance of the circuit. This observation suggests that a design system for integrated circuits should distinguish between wires and other components, and should treat wires as flexible connections of fixed width. A sketch is an abstraction of an integrated circuit layer that allows wires to be treated this way.

The type of system I envision would free the designer from concern with the geometry of wires. The designer would provide the system with rough routings for wires, and the system would either route them to form a legal layout, or else show the designer why no routing was possible. When the designer wished to move some of the circuit devices, the system would automatically bend wires and move other components as necessary to keep the layout legal. The problems of sketch routability, routing, and compaction embody the main computational tasks that such a system should be able to execute.*

Nature of this research

Despite its practical roots, this dissertation is in essence a theoretical study. I have not implemented any of my algorithms, nor are the sketch problems themselves designed to model the complex rules that IC designs must obey. Instead, the thesis is primarily concerned with the mathematical foundations of single-layer wire routing. My approach allows me, in the analysis of sketch problems, to trade complexity in the algorithms for complexity in their proofs of correctness. Thus the algorithms I present are relatively simple, but their justification occupies the bulk of this document.

Central to both the algorithms and the theory is the concept of a cut: a path in the routing region that spans two obstacles. As with other single-layer wire-routing problems, the congestions and capacities of the cuts in a sketch determine its routability. This theorem informs the algorithms for the sketch routability and compaction problems; its application to routability testing is evident. Taking the idea farther, I show that a sketch can be compacted by transforming the routability conditions given by cuts into constraints—simple linear inequalities—on the positions of obstacles. Solving the resulting constraint system reveals the optimal locations for the obstacles, including the terminals of wires. The compacted sketch can then be routed to restore the wires.

The sketch routing algorithm requires a deeper result, and a new concept: that

* Such a system has recently been implemented [36, 37]. Called 'Bubbleman', it employs similar ideas to those presented here, but its algorithms are quite different. Rather than solving global routing and routability problems, it incrementally builds a layout with minimal wire lengths as one inputs the components. It also performs two-dimensional compaction with automatic jog insertion via simulated annealing [19].

of a *half-cut*. Whereas a cut measures the congestion between two obstacles, a half-cut measures the congestion between an obstacle and a wire. Each half-cut for a wire constrains the routing of that wire: if the half-cut becomes too short, the other wires will be unable to fit across it. Certain of these constraints suffice to establish the optimal detailed routing of a sketch—the feasible realization whose wires have minimum length.

To a large degree, then, the study of single-layer wire routing is the study of cuts and half-cuts, and their interactions with wires. The subject has two parts: a mathematical part, which establishes the theorems concerning routability and minimum-length feasible realizations; and an algorithmic part, which concerns the computation of congestion for cuts and half-cuts, and the integration of this information over an entire sketch. Besides the division between algorithms and mathematics, there is another. The sketch compaction problem demands rather different techniques from the sketch routing and routability problems, and so I treat it separately.

Algorithmic ideas

Chapter 1 defines the sketch model and presents efficient algorithms for the sketch routability and routing problems. The idea behind these algorithms is the *conversion of topological conditions to geometric conditions.* My theory of single-layer wiring reduces the sketch routing and routability problems to two simpler problems:

(1) computing the congestion of a cut or half-cut, and

(2) finding the shortest routing of a wire that passes through certain line segments in a certain order.

Problem (2) happens to be equivalent to the task of finding the shortest path that passes in order through a sequence of triangles, each one sharing an edge with the preceding one. In this form the problem is evidently geometrical, and can be solved in linear time by a short algorithm. Problem (1) is harder.

To compute the congestion of a cut in a sketch, I use a data structure called the *rubber-band equivalent* of the sketch. This structure is built by shrinking every wire in the sketch to its minimum length. The shrunken wires, or *rubber bands*, make no more crossings with cuts than their topology dictates. Leaving aside some technical difficulties, the congestion of a cut is derivable from crossings it makes with rubber bands. The same goes for half-cuts. Thus the rubber-band equivalent expresses a topological quantity, congestion, in terms of a geometric quantity, a crossing number. Computational geometry provides the means to accelerate the computation of these crossing numbers. The construction of the rubber-band equivalent, too, is an essentially geometric process, and fairly efficient. In sum, geometric methods provide a conceptuall, uniform approach to sketch routing and routability testing, and lead to efficient algorithms for both problems.

Mathematical ideas

Chapter 2 begins a long technical development that culminates in correctness proofs for my sketch routing and routability testing algorithms. (Actually, some of the final steps are unfinished.) The theory revolves around a single concept: that of a simply connected *covering space* for the routing region. The covering space is a surface with infinitely many layers, each built from pieces of the routing region. The pieces are sewn together in such a way that every loop in the surface can be shrunk to a point. Paths in the covering space can be projected down to the routing region, and paths in the routing region can be *lifted* up into the covering space. Nearly every aspect of the theory exploits the special relationship between the multiply connected routing region and its simply connected covering space.

The covering space serves two primary functions. First, it provides a good definition of a *necessary* crossing between a cut and a wire. Informally, a necessary crossing is one that cannot be removed by rerouting the wire. The formal definition helps me to analyze the congestion of cuts, and to rigorously derive inequalities among the congestions of different cuts. Second, the covering space sorts out the interactions of *different parts of the same wire.* When a wire is lifted to the covering space, homotopically distinct parts of the wire fall on different layers. Thus the covering space transforms a problem with homotopy constraints (the rough routings) into a purely spatial problem. To show that a wire can be routed, I first find an appropriate routing within the covering space, and then project it to the routing region.

A second model

Unfortunately, the sketch model lends itself poorly to topological analysis: the covering space of the routing region does not permit lifting of wires and cuts. So my mathematical development employs a more elegant, but less practical, model, in which the analogue of a sketch is called a *design*. The design model supports a rich theory of routing that relates properties of cuts to the existence of various types of routings. It also identifies and characterizes the optimal, or *ideal*, routings of a routable design, and provides methods for computing the congestions of cuts and half-cuts. The design model differs sufficiently from the sketch model, however, that results in one cannot be applied directly to the other. Instead one must derive results concerning sketches by approximating the sketch with designs that, in some sense, converge to it.

I spend Chapters 4 through 7 exploring designs, Chapters 2 and 3 preparing for this exploration, and Chapter 8 applying the results to the sketch model. The chapter-by-chapter breakdown is as follows.

- Chapter 2 begins by stating many of the mathematical definitions and notations that will be used throughout the technical parts of the thesis. It also supplies a short introduction to homotopy theory and covering spaces, enough to appreciate the elementary ways in which I employ them. The rest of Chapter 2 claims, mostly without proof, theorems from topology that will be used sporadically in the following chapters.
- Chapter 3 defines the class of spaces, called *sheets*, that serve as the routing regions for designs. It then studies the topology of their simply connected covering spaces, which I call *blankets*, and of various sorts of paths in sheets and blankets. The central result is that when a *link* (e.g., a cut or wire) is lifted to a blanket, it separates the blanket into two pieces, a left side and a right side. This result allows us to recapture some of the simplicity of river routing in channels, where every cut and wire divides the channel. Thus Chapter 3 lays the real foundation for what is to come.
- Chapter 4 defines the design model and begins to develop the theory of cuts, half-cuts, and wires. It relates the congestion of a cut to the necessary crossings of the cut by wires, and it identifies congestion with a quantity called *flow* defined in terms of liftings to a blanket. Flow is a much more convenient and powerful concept than congestion, and much of Chapter 4 is concerned with relating the flows across different cuts. Not all the cuts we consider are straight; some even have self-intersections.
- Chapter 5 defines the ideal routings of the wires in a *safe* design, and proves that they form a valid routing of the design. The safety of a design is a function of its straight cuts, and primarily of the flows and capacities of those cuts. The result of the construction is that every safe design is routable. In more abstract terms, local routability implies global routability.
- Chapter 6 completes the proofs of the major theorems concerning designs. First it shows that unsafe designs are unroutable, providing a converse to the result of Chapter 5. It also shows that the arc length of ideal routings cannot be improved upon. Finally, it proves that the routability of a design depends only on the properties of a few straight cuts, not all of them. This observation makes it effective to test routability by testing safety.
- Chapter 7 goes on to consider techniques for routing and testing the routability of designs. The design model is ill-suited for the developement of routing algorithms, but the techniques developed with reference to designs can later be applied to sketches. Much of Chapter 7 revolves around the

use of rubber bands in routing and testing routability. It explains how to construct them, why they can be used to compute flow, and how they give rise to the structures (called *corridors* or *tunnels*) that we using in routing wires.

- Chapter 8 develops a careful correspondence between sketches and designs. It then shows how to use results in the design model designs to obtain results in the sketch model. Due to lack of time and space, some proofs are omitted. The outcome includes two major theorems concerning sketches, and also justifies my main algorithms for routing testing the routability of sketches.

I make no claims about the simplicity, shortness, or elegance of my proof techniques. Indeed, they could surely be improved, particularly if more advanced results in algebraic topology were assumed. They testify nonetheless that the tool of lifting to a simply connected covering space is apposite to wire routing with homotopy constraints. As long as my current proofs are, they might be even longer in another approach. For any approach must ultimately be based on a solid understanding of the role of homotopy in the wiring problem, such as I have tried to give in Chapters 3 and 4.

Compaction with flexible wires

Chapter 9 presents and proves correct a polynomial-time algorithm for sketch compaction. The algorithm requires a new approach to the manipulation of cuts because the geometry of a sketch can change radically during compaction. The topology of the sketch, on the other hand, is invariant. For this reason I introduce a second technique for computing congestions: a graph-theoretic method that works directly from the topology of the sketch. It includes an interesting preprocessing phase that speeds up searches through the graph that represents the sketch.

The chief difficulty in sketch compaction, however, is not in computing the congestions of cuts, but in deciding which cuts require consideration. As the obstacles in a sketch move, the relevant cuts change, as do their congestions. What the compaction algorithm actually examines is a set of *potential cuts*—cuts whose positions are functions of the configuration of the obstacles—that give rise to routability conditions. It turns out that by considering the potential cuts in a certain order, one can find for each potential cut the configurations in which it constrains the layout; the potential cut has the same congestion in all such configurations. So the compaction algorithm builds its constraint system iteratively, at each step considering the effects of adding a single potential cut. The algorithm itself is far from transparent; only in the analysis does its rationale become clear.

The analysis of the compaction algorithm leans heavily on the notion of a *configuration space.* In sketch compaction the configuration space is the vector space of possible displacements of the obstacles.* I relate the compaction algorithm to an *abstract algorithm* that manipulates subsets of the configuration space. (The actual, or *concrete*, algorithm represents these subsets by systems of linear inequalities.) I deduce the correctness of the abstract algorithm from four postulates concerning the sequence of potential cuts it evaluates. These postulates indeed hold for the potential cuts used by the compaction algorithm, and hence the correctness of that algorithm quickly follows. The advantage of this proof strategy is that changes in the model need not entail major changes in the proof; rather, it is enough to choose a sequence of potential cuts and check that they satisfy the postulates in the particular model one wishes to use.

Extensions and discussion

Chapter 10, the final chapter before the Conclusion, explores how far and how easily the sketch model can be extended in various ways. Among the possibilities it considers are these: allowing wires and obstacles to be made of different materials, each pair of materials with its own separation requirement; forcing wires to run in a grid; measuring the separation between wires with the euclidean metric; allowing wires to contain circular arcs; allowing obstacles to contain circular arcs; permitting the terminals of a wire to merge or pass through one another during compaction; routing with extended terminals, letting the points of connection move; and including wires with more than two terminals. In most cases the proposed changes in the sketch algorithms are relatively minor, but to justify them may be difficult (or even impossible, if one of my conjectures is false). The greatest problems arise in attempting to handle extended terminals and multiterminal nets. Chapter 10 proposes an alternative to the sketch model that, if it proves mathematically tractable, could eliminate these and several other problems.

The thesis concludes with a summary of its results, a comparison with some related work, a list of open problems, and several suggestions of directions for future research.

* The configuration space seems to be a natural tool for understanding compaction. It was by formulating the compaction problem in terms of configurations that I discovered a fact that could have significant practical consequences for compaction algorithms [30]. My observation, explained in Chapter 9, implies that Dijkstra's algorithm can be used to solve the standard one-dimensional compaction problem if the initial layout is legal.

Chapter 1

Sketch Algorithms

This chapter states precisely the major results of this dissertation. First it defines the sketch model and the problems of *sketch routability*, *sketch routing*, and *sketch compaction* that my algorithms solve. It then considers a data structure for a sketch, called its *rubber-band equivalent*, which supports computation involving the sketch topology, and thereby speeds up the algorithms for sketch routability and sketch routing. Next it presents algorithms for those two problems. Both algorithms have worst-case running time $O(n^2 \log n)$ on input of size n. Then, in Sections 1E and 1F, I show that the performance of these algorithms is limited mainly by the routability testing procedure, and I present several methods for improving its average-case running time. By exploiting the idea of *shadowing* [6] we obtain an algorithm for sketch routability that runs in time $O(n^{3/2} \log n)$ on the average. All proofs are deferred to Chapter 8. Sections 1A through 1D represent joint work with Charles Leiserson.

1A. The Sketch Model

We begin by defining sketches and the natural problems set in that model. This section also states the sketch routability theorem and the sketch routing theorem in which my algorithms are grounded.

A sketch is an abstraction of the wiring on a single layer of an integrated circuit or printed circuit board. It represents the topology and the geometry of that layer, but none of its electrical or functional characteristics. I have chosen the sketch model for its simplicity, its similarity to existing theoretical models, and its ease of implementation. Consequently, it deals only with piecewise linear objects. This limitation is not serious, for in practice one often approximates curved structures by polygons in order to avoid the problems of computing with irrational numbers (in whatever representation). A more serious drawback is that sketches, as described here, cannot satisfactorily represent wires with more than two terminals. These issues are discussed further in Chapter 10.

Since a sketch must distinguish between flexible and rigid objects, it has two types of components: *traces*, which represent either rough routings or detailed routings of wires, and *features*, which represent terminals, devices, and routing obstacles. A **feature** is a point or line segment in the plane, and a **trace** is a piecewise linear path with the following properties.

(1) The path has no self-intersections.

(2) The path touches no features except at its endpoints.

(3) The endpoints of the path are features—the **terminals** of the trace.

(4) Each terminal is a point, isolated from the other features.

A **sketch** is a finite set of features that intersect only at their endpoints, together with a finite set of nonintersecting traces. The connected groups of features in a sketch are called **islands**. By (3), every terminal is an island; the islands that are not terminals are called **obstacles**. The **routing region** of a sketch is the set of points that lie on no feature. Islands and traces are collectively called **elements**.

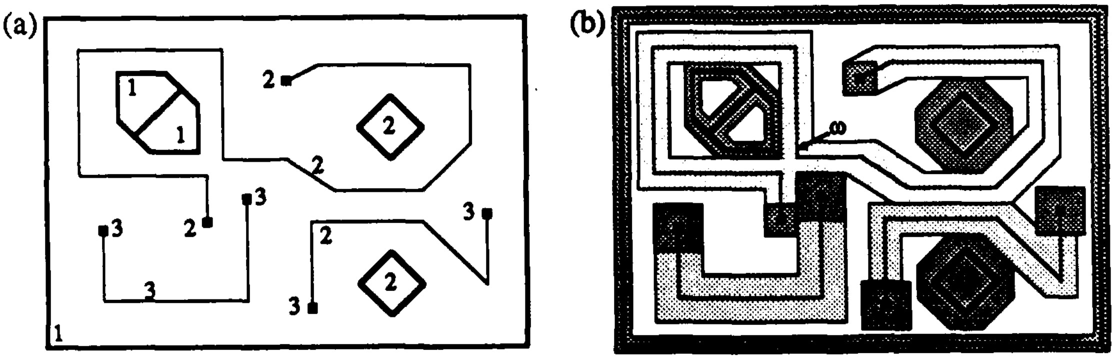


Figure 1a-1. ***A typical sketch and the territories of its elements.*** **Part (a) illustrates a simple sketch. Dark line segments are features, light paths are traces, and the number nearest to each element indicates its width. Part (b) shows each element's *territory*, which takes its width into account. The territory of a trace is not shown where it overlaps the territories of its terminals.**

The elements of a sketch actually represent the centerlines of regions in the wiring layer. Hence we associate with each element a positive number called its **width** that indicates how much space it actually requires. No trace may have greater width than either of its terminals. The **territory** of an element of width d is the set of points whose distance from that element is less than $d/2$.

We measure distance using a piecewise linear norm,[*] denoted $\|\cdot\|$, that is the same for all elements. I call this norm the **wiring norm**, because different norms

[*] See the glossary for an explanation of wiring norms. The examples in this chapter

give rise to different wiring models. Terms like 'distance' and 'closest'—but not 'arc length'—refer to measurement in the wiring norm unless otherwise specified. The distance in the wiring norm between two points p and q is $\|p - q\|$, and the distance between two regions P and Q is

$$\|P - Q\| = \inf_{p \in P} \inf_{q \in Q} \|p - q\| .$$

Depending on the placement of its elements, a sketch may or may not represent a valid circuit layout. If it does, the sketch is called **proper**. In my model a sketch is proper if the elements that should not interact are properly separated. Two elements are assumed to interact if and only if their territories overlap. Sometimes this interaction is good, as when a trace connects to its terminals. Thus we consider a sketch to be improper if it has two elements with overlapping territories, unless those elements are a trace and one of its terminals.

There is one further constraint on proper sketches. It arises because a trace must be separated from *itself*, lest it form a loop in the layout. Let us say that a trace is **self-avoiding** if the set of points lying outside its territory and outside the territories of its terminals has only one connected component that includes islands of the sketch. In other words, the territory of a self-avoiding trace, together with those of its terminals, does not separate any two islands from one another. All the traces in a proper sketch must be self-avoiding. The sketch in Figure 1a-1 fails to be proper because the trace ω is not self-avoiding.

Sketch routing problems

The single-layer routing problems I consider take a sketch as input. This sketch is not expected to be proper. Instead, each trace in the input sketch represents a rough routing; it defines a set of possible realizations for that trace. A realization of a *sketch* is a sketch that results from routing each of its traces, that is, replacing them by realizations. We say that a sketch is **routable** if it has a proper realization. The **sketch routability problem**, then, is just the problem of determining whether a sketch is routable. It turns out that whenever a sketch is routable, it has a proper realization that simultaneously *minimizes the length of every trace*. The **sketch routing problem** is to find this realization if it exists.

To route a trace in a sketch, one deforms the trace in a continuous fashion. The notion of continuous deformation is made precise as follows. We define a **bridge** to be a piecewise linear path in a sketch that intersects features of the sketch at its endpoints only. Then all traces are bridges. We think of a bridge as a continuous

use the L^∞ norm, in which the distance between two points is the maximum of their horizontal and vertical separation.

function from the unit interval $I = [0,1]$ to the plane R^2. Two traces of the same width, say θ_0 and θ_1, are **bridge-homotopic** if they are part of some family of bridges $\{\theta_t : t \in I\}$ such that the function $T\colon I \times I \to R^2$ defined by $T(s,t) = \theta_t(s)$ is continuous and piecewise linear. The function T is a *homotopy* or "continuous deformation" of bridges. If θ is a trace in a sketch S, then a **route** for θ is any bridge that is bridge-homotopic to θ in S. A **realization** of θ is a *trace* that is a route for θ. The realization is **feasible** if it is part of a proper realization of the sketch S.

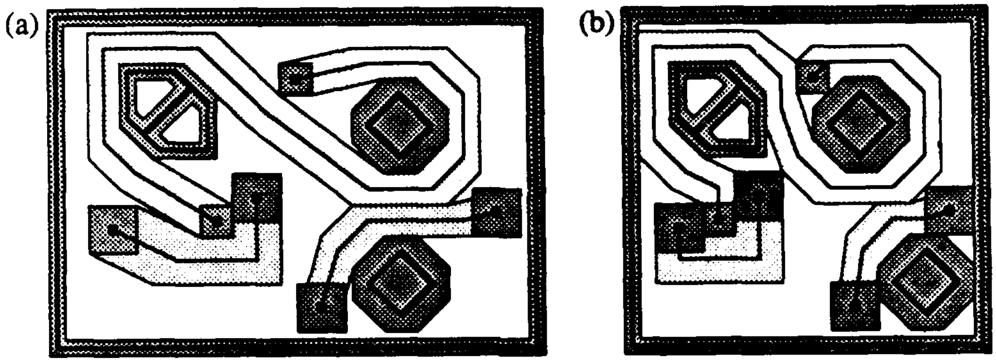


Figure 1a-2. *A proper realization of a sketch and a compacted version of it.* Part (a) is a proper realization of the sketch in *Figure* 1a-1: the territories of its elements are disjoint, except where traces contact their terminals; and every trace is self-avoiding. (Territories are open sets; they do not include their boundaries.) Every trace in this realization has minimum length. Part (b) shows a compacted version of this sketch. If we allow the islands to move sideways independently, then among all the proper sketches that are reachable from the configuration at left, the sketch in (b) has minimum width.

The **sketch compaction problem** is a generalization of the sketch routing problem that involves moving features as well as traces. The input to this problem is a routable sketch with the islands grouped into **modules**; each module is allowed to move horizontally as a unit. Modules may not move vertically. As modules move traces must move as well in order to remain connected to their terminals. Let us say that a sketch is **reachable** if it can be obtained from the input sketch by a continuous, piecewise linear motion that maintains the routability of the sketch. (The motion of each trace should be a piecewise linear homotopy, though not one that necessarily fixes its endpoints.) The sketch compaction problem is to find a proper, reachable sketch of minimum width. Solving this problem allows one to perform one-dimensional compaction of VLSI layouts, inserting jogs into wires automatically; a special case of this problem was considered in [29].

Major results

This thesis presents polynomial-time algorithms for the sketch routability, routing, and compaction problems. Given as input a sketch of size n, the routing and routability testing algorithms run in time $O(n^2 \log n)$, while the sketch compaction algorithm runs in time $O(n^4)$. All are fairly easy to implement, and are efficient enough to be useful in practice.

The correctness of these algorithms rests on the theory of single-layer wiring. This theory gives necessary and sufficient conditions for a sketch to be routable, and provides methods for testing these conditions. For routable sketches, it also characterizes the minimum-length feasible realizations of traces. The tools of the theory are the techniques of point-set and algebraic topology; the objects it studies are traces and *cuts*.

One important result of the theory says, in essence, that a sketch is unroutable if and only if too many traces are forced to pass through the "channel" between some pair of islands. This statement may seem obvious, but it is far from trivial. We formalize it using the idea of a cut. A line segment is a **cut** of a sketch if it touches the features of the sketch at its endpoints only. (More properly, a cut is a linear path, and we write the cut $\overline{pq}$ as $p \triangleright q$ if we wish to emphasize its orientation from p to q.) Each cut has a **capacity** that represents the maximum total width of the traces that can cross it. If endpoints of the cut $\overline{pq}$ lie on the islands P and Q, then we define

$$\text{capacity of } \overline{pq} = \text{length of } \overline{pq} - (\text{width of } P)/2 - (\text{width of } Q)/2.$$

The length of $\overline{pq}$ is measured in the norm used to define territories.

Each cut also has a **congestion** that measures the total width of the traces forced to pass across it. To define it, we first define the **entanglement** of a trace with a cut $\overline{pq}$ to be the minimum number of crossings of $\overline{pq}$ by any route for the trace. Crossings that occur at p or q do not count. The entanglement of a trace with a cut represents, in some sense, the number of **necessary crossings** of the cut by the trace. Intuitively, a necessary crossing is one that cannot be removed by applying a bridge homotopy to the trace. This intuitive notion is not easy to formalize, however, so we leave it informal until Section 4B. Congestion is defined in terms of entanglement. If Θ denotes the set of traces in the sketch, then we define

$$\text{congestion of } \overline{pq} = \sum_{\theta \in \Theta} (\text{width of } \theta) \cdot (\text{entanglement of } \theta \text{ with } \overline{pq}).$$

If the congestion of a cut exceeds its capacity, then the traces will not be able to fit across the cut. We say a cut is **unsafe** if its congestion exceeds its capacity. This does not always mean the sketch is unroutable, however, because there may

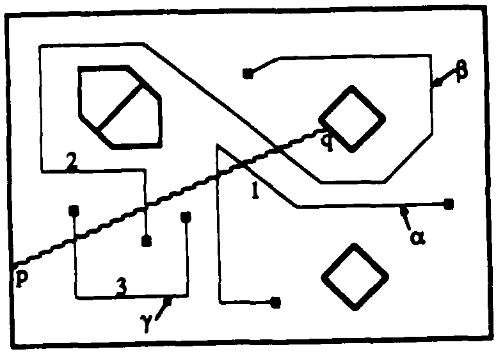


Figure 1a-3. *The attributes of cuts.* In the sketch depicted here, the dashed line $\overline{pq}$ is a nonempty cut. Three traces intersect this cut. The trace α has width 1 but entanglement 0 with $\overline{pq}$, as the crossings it makes with $\overline{pq}$ are unnecessary. The trace β has width 2 and entanglement 2, and the trace γ has width 3 and entanglement 1. Hence the congestion of the cut $\overline{pq}$ is 7. If its capacity is 7 or greater, this cut is safe.

not *be* any traces crossing that cut. A cut is **empty** if it has zero congestion and its endpoints lie on the same island. Empty cuts have no bearing on routability. A nonempty, unsafe cut, on the other hand, means that the channel it spans is congested. A sketch is **safe** if and only if all its nonempty cuts are safe.

Now the theorem concerning routability can be stated more precisely. If a sketch is safe, then it is routable. Conversely, every routable sketch is safe. I call this result the **sketch routability theorem**. It suggests that the routability of a sketch may be checked by testing whether certain cuts of the sketch are safe. One can easily find a small set of *critical* cuts with the property that if any nonempty cut is unsafe, some nonempty critical cut is unsafe. My algorithm for the sketch routability problem works by testing the safety and emptiness of these critical cuts.

More significant than the sketch routability theorem, however, is the **sketch routing theorem**, which yields minimum-length feasible realizations for the traces in a routable sketch. This theorem cannot be fully stated here, because it depends upon a complicated construction of traces called *ideal realizations*. Intuitively, an ideal realization of a trace in a routable sketch is a minimum-length route for that trace that stays far enough away from the islands to permit the other traces to be routed. Every trace in a routable sketch has a unique ideal realization. The sketch routability theorem states two things. First, if every trace in a routable sketch is replaced by its ideal realization, then the resulting sketch is proper. Second, no shorter feasible realizations exist for those traces. To solve the sketch routing problem, therefore, one need only be able to compute the ideal realization of each trace in a routable sketch. My routing algorithm does just this.

1B. The Rubber-Band Equivalent of a Sketch

My algorithms for sketch routability and routing both rely on a data structure called the **rubber-band equivalent** (RBE) of the sketch. This structure solves the central difficulty associated with the processing of sketches, namely the integration of geometric and topological information. Methods from computational geometry can be applied to the RBE to compute the congestions of cuts and to find constraints on the positioning of traces. In this section I define the RBE, show how to construct it, and explain the operations that it supports.

I assume the input sketch is represented as a pair of data structures: a set F of features, and a set T of traces. Let us denote the size of a data structure D by the symbol $|D|$. Each feature is a point or line segment, and hence requires constant space to represent. Each trace, being piecewise linear, is represented as a sequence of line segments. Thus $|F|$ is proportional to the number of features in F, and $|T|$ is proportional to the number of line segments that compose the traces in T. If S is the sketch (F, T), then we have $|S| = |F| + |T|$. The algorithm given in this section computes the rubber-band equivalent of a sketch $S = (F, T)$ in time $O(|F|\,|T| \log |S|)$.

Motivation

Intuition suggests that if a trace crosses a cut more times than necessary, then it contains an unnecessary detour. If we could make each trace as short as possible, then the number of crossings between a cut and a trace would equal their entanglement. Unfortunately, most traces have no minimum-length routes, for a trace is not permitted to contact any features but its terminals. So we construct instead the **rubber band** of each trace: the shortest path, in euclidean arc length, that is the limit of a sequence of routes for that trace. Intuitively, we shrink the trace to its minimum length, allowing it to touch features but not to cross over them. The resulting path is a sequence of line segments whose endpoints are feature endpoints.

If we replace every trace in a sketch by its rubber band, the result is not, in general, a sketch. It nevertheless can be treated as a sketch in which features and traces have infinitesimal separation. Wherever a rubber band touches a feature, we consider it to leave the feature to its left, leave the feature to its right, or else connect to the feature (if the feature is one of its terminals). Similarly, wherever one rubber band touches another, the second rubber band falls either left or right of the first. No rubber band ever crosses over another one, and hence this adjacency information can be assigned in a consistent manner to all the features and rubber bands. The RBE of the sketch stores this information in a concise form.

The RBE helps one to compute, for any desired straight cut, the sequence of traces that necessarily cross it, in order along the cut. This sequence is called the **content** of the cut. (It may contain the same trace more than once.) The content of a cut nearly equals the sequence of rubber bands that cross the cut, the difference being that one sequence consists of traces while in the other one consists of the corresponding rubber bands. The tricky part, of course, is defining

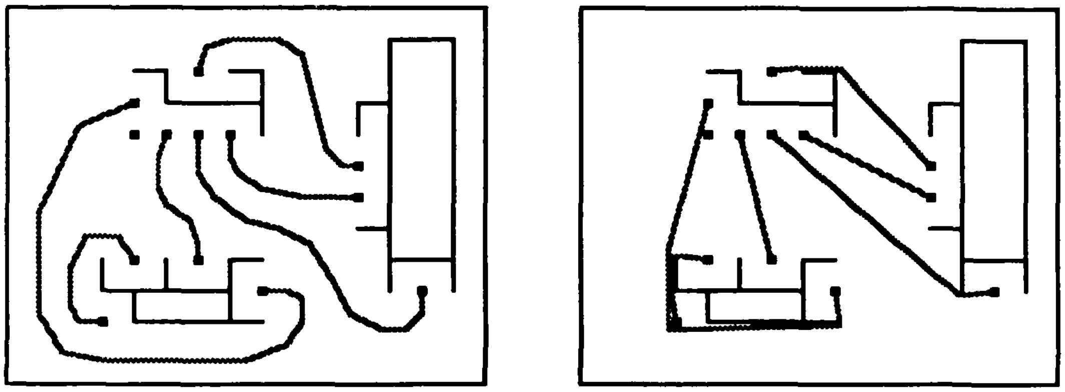


Figure 1b-1. *The rubber-band equivalent of a sketch.* The sketch is on the left, its RBE on the right. In the RBE, features and rubber bands that are shown here as adjacent segments actually overlap. The strands have been artificially displaced to show the adjacency relations among the features and rubber bands.

which rubber bands cross a cut and in what order they do so. Here the adjacency information comes in. Some places where the cut intersects a rubber band should not be considered crossings. For example, if the cut intersects a feature from the top, and the rubber band runs along the bottom of the feature, one should think of them as being separated by an infinitesimal distance. If one filters out such intersections, the remaining ones correspond exactly with the traces in the cut's content. Moreover, the cut can be considered to cross the rubber bands in a certain order, because even where the rubber bands overlap, their adjacency relation orders them totally. This ordering is irrelevant for computing flow but highly significant for wire routing, as explained in Section 1D.

Definition and use of the RBE

The RBE of a sketch is essentially a planar multigraph with some extra structure. Its nodes are feature endpoints; its arcs are features and **cables**, which are groups of rubber band segments. For each pair $\{p, q\}$ of feature endpoints there can be up to three cables from p to q: one on each side of the cut or feature $\overline{pq}$, and one that crosses over the cut $\overline{pq}$. The rubber band segments within each cable are called its **strands**, and are totally ordered. We represent the ordering by means of a height-balanced tree. In addition, the features and cables radiating from each feature endpoint are circularly ordered as shown in Figures 1b-2 and 1b-3. We store this ordering in a pair of height-balanced trees by breaking it into total orderings as explained later. In effect, these orderings specify which features and strands would be adjacent if the rubber bands had infinitesimal thickness. The total orderings within cables, combined with the circular ordering on the cables that touch a feature endpoint, give rise to a circular ordering on the *strands* that touch a feature

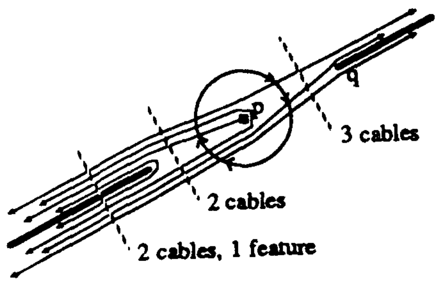


Figure 1b-2. *The circular ordering of cables at a feature endpoint.* The arrows depict the circular ordering of cables at feature endpoints p and q. There can be up to three cables having p and q as endpoints.

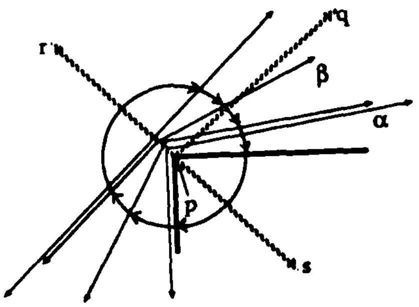


Figure 1b-3. A crossing sequence. The crossing sequence of the ray $\overrightarrow{pq}$ emanating from p consists of the rubber bands in cable α followed by the rubber bands in cable β.

endpoint.

The RBE supports the following operation: given a ray emanating from the feature point p, report its **crossing sequence at p**: the sequence of rubber bands that *cross over* that ray at p. Rubber bands that end at p are not part of the crossing sequence, nor are rubber bands that are parallel to the ray at p. The content of a cut $p \triangleright q$, converted into a sequence of rubber bands, is then the concatenation of three lists.

(1) First is the crossing sequence of the ray $\overrightarrow{pq}$ at p.

(2) Next come the rubber bands whose strands cross of the middle of $p \triangleright q$. The strands are sorted by distance from p, and ordered within each cable as well.

(3) Last is the reverse of the crossing sequence of $\overrightarrow{qp}$ at q.

If p is a point on a feature but not a feature endpoint, then the crossing sequence of a ray at p can be computed without new data structures. Let $\overline{qr}$ be the feature containing p. The crossing sequence of a ray $\overrightarrow{ps}$ is just the sequence of strands in the cable (if any) lying along $\overline{qr}$ on the same side as s. This cable can be found by examining the circular order at q or r, because if it exists, it must be adjacent to the feature $\overline{qr}$.

To compute crossing sequences at a feature endpoint p we use one of two different data structures. If all the cables touching p fall on a line ℓ, as shown in Figure 1b-2, then it suffices to store four different crossing lists: two for the rays lying in ℓ, and two for rays pointing into the half-planes of ℓ. In this case, at most six arcs (features and cables) connect to the node p, so their circular ordering can be represented by a constant-size data structure. If the cables touching p are not all parallel, on the other hand, then we have the situation of Figure 1b-3. There is a ray $\overrightarrow{pr}$ whose crossing sequence at p is longest, and a ray $\overrightarrow{ps}$ whose crossing sequence at p is empty. Moreover, the crossing sequence at p of an intermediate ray $\overrightarrow{pq}$ is obtained by enumerating the strands in the cables interior the angle $\angle spq$, where the interior of this angle is chosen not to include the ray $\overrightarrow{pr}$. To enumerate these cables quickly, we break the circular ordering of the arcs incident on p into two total orderings. The arcs between $\overrightarrow{pr}$ and $\overrightarrow{ps}$ clockwise are stored in one height-balanced tree, and the arcs between $\overrightarrow{pr}$ and $\overrightarrow{ps}$ counterclockwise are stored in another height-balanced tree.

Constructing the RBE

The rubber-band equivalent of a sketch can be computed fairly efficiently. First one triangulates the routing region with cuts, which I call **doorways** or simply **doors**. This operation is efficient—it requires only $O(|F|\log|F|)$ time—and fairly standard in computational geometry [43], so I shall not dwell on it. Next one constructs the planar graph whose nodes are feature endpoints and whose arcs are feature segments. Nodes can be represented initially without using height-balanced trees, but when cables are added to the graph, some nodes will have to be converted to the more general data structure. Then comes the interesting part: for each trace in the sketch one computes its rubber band and inserts it into the data structure. I describe the insertion operation first.

Given the rubber band of a trace, one inserts its strands in order. To insert the strand $\overline{qr}$, first determine which cable it belongs in. There can be up to three cables from q to r: one to the left of the ray $\overrightarrow{qr}$; one in the middle, which crosses over $\overrightarrow{qr}$; and one to the right of $\overrightarrow{qr}$. If the new strand leaves q and r to different sides, it goes in the middle cable. Otherwise if it leaves either q or r to the left or right, it goes in the left-hand or right-hand cable, respectively. If this strand is the entire rubber band, so that it has q and r as terminals, it goes in the left-hand cable by default. If the appropriate cable for the strand $\overline{qr}$ does not exist, create it and insert it into the circular orders at q and r.

The case in which the cable exists is more difficult. If $\overline{qr}$ is the first strand in its rubber-band (i.e., q is its terminal), then insert it at the right-hand edge of the left-hand cable or the left-hand edge of the right-hand cable, as appropriate. Otherwise let $\overline{pq}$ be the strand preceding $\overline{qr}$ in its rubber band, and find which strands are adjacent to $\overline{pq}$ in the circular order at q, ignoring those that connect to q as a terminal One of these is connected to a strand X that goes to r. Insert $\overline{qr}$ adjacent to X.

Making rubber bands

To find the rubber band for a trace θ, we follow θ through the triangulation, and record the sequence of doorways that θ passes through. When θ crosses a doorway $\overline{pq}$ but immediately returns, the doorway $\overline{pq}$ may be removed from the sequence, because it represents an unnecessary detour. After eliminating such unnecessary doorways, which one can do in linear time, one is left with the sequence of doorways that the rubber band for θ passes through. Let us call this sequence of doorways a **corridor**. The shortest path through this corridor that connects the terminals of θ is the rubber band for θ.

I now outline a linear-time algorithm to find the shortest path through a corridor. Each door in a corridor may share an endpoint with the previous door (or with the first terminal of the wire, if this is the first door), and hence has either one or two new vertices. We represent a corridor as the sequence of new vertices, together with an indication of which vertices lie to the left of the path, and which lie to the right. The algorithm examines the vertices one by one, keeping track of left and right boundaries for the shortest path. Suppose that a new vertex of the nth doorway, call it l, lies to the left of the path, and let t denote the initial terminal of the trace. After examining l, the left boundary is the shortest path through the first $n-1$ doorways from t to l. Similarly, after examining a right vertex r, the right boundary is the shortest path in the corridor from t to r. The boundaries are piecewise linear paths, stored as sequences of vertices.

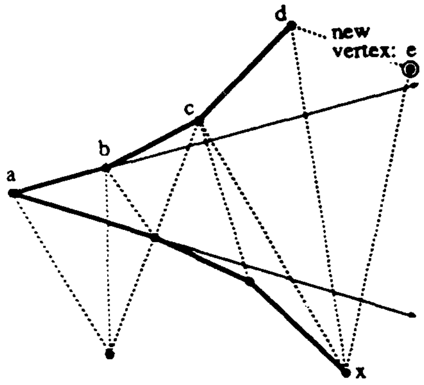


Figure 1b-4. *A snapshot of Algorithm W.* Only active vertices are shown. Dotten lines are the doors of the corridor, dark lines are boundaries for the optimal path, and light lines are rays of visibility. If e is a left vertex, the algorithm will remove the points c and d from the left boundary in favor of e. If e is a right vertex, it will replace the right boundary from a to x by the segments $\overline{ab}$ and $\overline{be}$.

Simple visibility tests are used to maintain the boundaries. The vertex at which the left and right boundaries diverge, and all vertices following it, are called **active**. When a left vertex l is encountered, the algorithm finds the oldest active left vertex v whose line of sight to l falls right of the left boundary. Next, it removes the portion of the left boundary that follows v. If the right boundary blocks v from seeing l, then the left boundary is extended along the right boundary until l is visible from the end of the left boundary. Finally, the point l is added to the left boundary. Symmetrical actions occur upon examination of a right vertex.

Algorithm W, shown below, is a linear-time implementation of this path-finding procedure. It uses stacks to represent the boundaries, and employs two simple geometric tests to maintain them. The function R-TURN(p,q,r) determines whether the point r lies to the right of the ray $\overrightarrow{pq}$; similarly, L-TURN(p,q,r) is true when r lies to the left of $\overrightarrow{pq}$. The algorithm assumes that consecutive doors are not collinear, and that the corridor contains the final terminal of the trace as both left and right vertices.

Algorithm W. (Finds a minimum-length path through a corridor.)
Input: Corridor vertices $C[1..n]$; initial terminal t.
Local variables: arrays of points $L[1..n]$ and $R[1..n]$; integers b, i, l, and r.
Output: the vertices $L[1..l]$ of a piecewise linear path.

```
1. $l, r, b \leftarrow 1$; $L[l], R[r] \leftarrow t$;
2. for $i \leftarrow 1$ to $n$ do
3.   if $C[i]$ is a left vertex then
4.     while $l > b$ and R-TURN$(L[l-1], L[l], C[i])$
5.       do $l \leftarrow l - 1$;
6.     while $r > b$ and not L-TURN$(R[b], R[b+1], C[i])$
7.       do $b \leftarrow b + 1$; $l \leftarrow l + 1$; $L[b] \leftarrow R[b]$;
8.     $l \leftarrow l + 1$; $L[l] \leftarrow C[i]$
9.   else (copy lines 4–8, exchanging $L$, $l$, and L-TURN for $R$, $r$, and R-TURN).
```

One can extend Algorithm W to determine, for each feature endpoint that the output path passes over, on which side of the path it lies. Proving the correctness of Algorithm W is straightforward.

Complexity analysis

The time and space performance of the RBE construction are dominated by the processing of strands. Each trace segment passes through $O(|F|)$ triangles, and hence gives rise to $O(|F|)$ strands. Hence the number of strands in the RBE is $O(|F|\,|T|)$, and this bound is tight in the worst case. In practice the number should ordinarily be much smaller. Algorithm W generates each strand in $O(1)$ time, and a strand can be inserted into the RBE in time $O(\log |S|)$. (The log factor derives from the use of height-balanced trees.) Therefore the construction of the RBE requires time $O(|F|\,|T| \log |S|)$ and space proportional to the size of the output, namely $O(|F|\,|T|)$.

Given a feature and a ray beginning on that feature, the RBE can produce the crossing sequence of that ray in time proportional to its length. This performance is optimal for the purposes of my routing algorithm, but not for my routability testing algorithm. To find the congestion of a cut $\overline{pq}$ one need not compute its content or the necessary crossings of that cut by traces. One need only compute the sum of the widths of the traces in the content of $\overline{pq}$. (Traces that appear more than once in the content are counted according to multiplicity.) Hence for the purpose of routability testing a condensed form of the RBE is needed. In this data structure, the strands within each cable are not distinguished; instead each cable is assigned a **width** that represents the sum of the widths of its strands. The **condensed RBE** also stores the width of every possible crossing sequence a ray could have; this requires storing 2 numbers per feature segment and at most $2n$ numbers at each vertex of degree n. These values can be computed in linear time from the widths of the cables, and the correct value for a ray can be found in $O(\log |S|)$ time.

Thus the **condensed RBE** is a planar multigraph whose vertices are feature endpoints and whose edges are cables. In this graph, at most three edges connect each pair of vertices—three cables, or one feature and two cables. Since the number of edges in a planar graph is at most linear in the number of vertices, the condensed RBE uses only $O(|F|)$ space. The workspace needed for its construction is also $O(|F|)$.

1C. Testing the Routability of a Sketch

A corollary to the sketch routability theorem shows that a sketch is routable if and only if its nonempty **critical** cuts are safe. We say that the critical cuts are **decisive**, because their safety and emptiness decide the routability of the sketch. A critical cut is a cut that begins at a feature endpoint and travels to the closest point on another feature. The distance is measured in the wiring norm; ties are broken using the euclidean metric. For any reasonable wiring norm, the critical cuts can be easily identified; there are $O(|F|^2)$ of them.

Thus the problem of routability testing is reduced to the problem of checking the safety and emptiness of a cut. For each critical cut, we need to know its congestion, its capacity, and whether its endpoints lie on the same island. The last condition is easy to test, because the islands of a sketch can be determined in $O(|F|^2)$ time. The capacity of a cut is also easy to compute, for it depends only on the distance between the cut's endpoints. I assume that the wiring norm of a vector can be computed in constant time. The congestion, on the other hand, is relatively hard to compute; for this we use the rubber-band equivalent of the sketch. This section presents an algorithm to test the routability of a sketch in time $O(|F|^2 \log |F|)$, given its condensed RBE.

The scanning technique

To check each cut quickly, we use an idea from computational geometry called **scanning**. This technique involves sweeping a scan line across the plane, while keeping track of the objects that intersect the line. The data structure representing those objects can then facilitate the computation of geometric quantities such as the congestion of a cut. If that data structure can be updated and queried quickly, it speeds up the algorithm by eliminating repetitive access to the objects being examined. An **event list** drives the scanning process by specifying the order in which objects enter and leave the data structure, and when the structure should be queried. The algorithm constructs the event list before scanning, and simulates the motion of the scan line by processing the events in order.

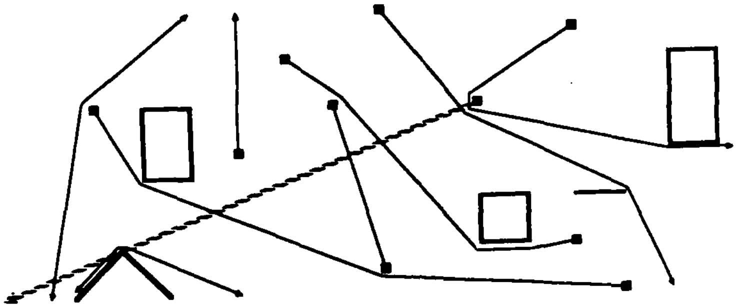


Figure 1c-1. *A snapshot of Algorithm T.* Here the algorithm is shown checking the safety of the critical cuts that begin at p. Algorithm T simulates the motion of a ray that sweeps around p like a radar beam. It uses data structures that support fast insertion, deletion, and search to represent the features and cables intersecting the scan ray. Whenever the scan ray includes a possible critical cut, Algorithm T can quickly determine whether this line segment is a cut, and if so, quickly compute its congestion.

For testing routability, the scan line will be a ray emanating from a feature endpoint p. As the ray sweeps through the RBE, it will occasionally intersect another point q such that $\overline{pq}$ is a critical cut. When this happens, the algorithm computes the congestion and capacity of $\overline{pq}$. If the cut is nonempty and the congestion is greater than the capacity, then the sketch is unroutable. The congestion is the sum of three quantities. The first is the total width of the cable strands that intersect the scan ray strictly between p and q; it is obtained from the data structure representing the scan ray. The other two are the total widths of the crossing sequences of the rays $\overrightarrow{pq}$ at p and $\overrightarrow{qp}$ at q; they are provided by the RBE. To construct the event list for scanning, the algorithm sorts all the relevant points in the RBE by angle as seen from p. Objects that touch p are left out. Both the sorting and the data structure accesses require only $O(\log|S|)$ time per object. Since there are $O(|F|)$ feature endpoints to scan from, the total execution time is $O(|F|^2 \log|S|)$.

Details of Algorithm T

The condensed rubber-band equivalent of a sketch consists of two types of line segments: features and cables. To store the objects crossing the scan ray, therefore, our algorithm uses two dynamic sets, FS and WS. The FS data structure contains features, while WS is a set of cables, each of which is weighted according to the total width of the strands in the cable. The operations on FS are insertion, deletion, and

- MIN?(s), which returns *true* if s is the nearest segment in FS to the origin p, and otherwise *false*.

If the scan ray intersects s at q, then MIN?(s) determines whether $\overline{pq}$ is a cut. The set WS supports insertion and deletion of cables, plus

- WIDTH(s), which returns the total width of the cables in WS lying strictly between the query segment s and the origin p.

If some cable stretches from p to q, then WIDTH(s) returns the width of the cable (if any) that crosses over the cut $\overline{pq}$.

The set FS is easily implemented so that each operation runs in $O(\log|F|)$ time by using a height-balanced search tree, sorted by distance from p. When two segments touch at their endpoints, their order in FS is unimportant, and so the closest segment to p can always be defined. Since features never cross, the order of segments within the set does not change. To execute MIN?(s), first query the condensed RBE to determine whether p is connected to a feature in the direction of the scan ray. Return *false* if so, and otherwise return *true* if and only if s is the first (leftmost) element of FS.

The structure WS can also be a height-balanced search tree. Since the number of cables in the condensed RBE is $O(|F|)$, each operation on WS will take $O(\log|F|)$ time. The WIDTH operation can be implemented by storing in each node the total width of the cables in its left subtree, plus the width of the cable stored in the node itself. These values are easy to maintain under the standard tree-balancing operations. The value WIDTH(s) can then be found by searching the tree for the farthest segment that is strictly closer than s. Every time the search path branches right or stops at a node, accumulate the quantity in that node. If the result is positive, it is WIDTH(s). Otherwise let q be the point at which the scan ray intersects s, and query the RBE for the width of the cable (if any) from p to q that crosses over the cut $\overline{pq}$. Take the result to be WIDTH(s).

The event list for scanning around a point p consists of two types of events. First, there is an event for every endpoint of a feature or cable in the RBE, except those objects that intersect p. A point may correspond to more than one event if two or more objects intersect there. Second, for each feature f there is an event corresponding to a point q on f such that $\|p-q\|$ is critical. The line segment $\overline{pq}$

is called a **possible** critical cut; it is a critical cut unless it crosses a feature. The points of the event list are sorted by angle as seen from p; angles are measured with respect to some reference ray. Events at the same angle may be sorted arbitrarily. Each point in the list is marked with a pointer to its segment, and whether it is the first point of its segment, the closest point, or the last point. If the segment is seen edge-on, either point can be "first".

Algorithm T. (Tests the routability of a sketch.)
Input: The condensed rubber-band equivalent of a sketch (F, T).
Local variables: Data structures FS and WS; event list EL; points p and q; feature f; cable c; congestion value t.
Output: Either *true* (routable) or *false* (unroutable).

```
Group the features into islands;
foreach feature endpoint $p$ do
   Initialize FS and WS to represent the reference ray;
   Clear EL;
   foreach feature $f$ do
      if $f$ does not touch $p$ then add events to EL for $f$;
   foreach cable $c$ do
      if $c$ does not touch $p$ then add events to EL for $c$;
   Sort EL;
   foreach event $e \in$ EL (in sorted order) do
      Update FS and WS;
      if $e$ is the possible cut $\overline{pq}$ to feature $f$ and MIN?($f$) then
         $c \leftarrow$ WIDTH($f$) + width of crossing sequences of $\overline{pq}$ at $p$ and $\overline{qp}$ at $q$;
         if $c > 0$ or $p$ and $q$ lie in different islands then
            if $c >$ capacity of $\overline{pq}$ then return false;
return true.
```

The operation of the algorithm is simple. It first finds the islands of the sketch by checking which features intersect which others. This takes at most $O(|F|^2)$ time. Then, for each feature endpoint p, it initializes the data structures FS and WS, constructs the event list for p, and simply scans through it, taking appropriate actions for each event. If the event is the first or last one involving that segment, the segment is inserted or removed, respectively, from the appropriate set. If the event is the closest point q on a feature f, and MIN?(f) is true, then the algorithm computes the congestion of the cut $\overline{pq}$ and checks whether this cut is empty. If not, and if its congestion is greater than the capacity of $\overline{pq}$, the algorithm signals that the sketch is unroutable.

1D. Routing a Sketch

This section presents a polynomial-time algorithm for producing a proper routing of a sketch, given its rubber-band equivalent. The algorithm minimizes the length of every trace in the routing, so that total trace length and the length of the longest trace are simultaneously optimized. To process the sketch $S = (F, T)$ the algorithm uses $O(|F|\,|T|)$ space and $O(|F|\,|T| \log |S|)$ time; these bounds are nearly optimal, for the output sketch may contain $\Omega(|F|\,|T|)$ trace segments. The output is guaranteed to be a proper routing if one exists, but otherwise it need not even be a sketch; it may contain illegal intersections. Hence the sketch to be routed should first be tested for routability using Algorithm T of the previous section.

The routing strategy

The routing algorithm examines the necessary crossings of cuts to generate constraints on the output traces. Every cut has a content, the sequence of traces that it necessarily crosses. Suppose that the cut $p \triangleright q$ from the island P to the island Q has content $\langle \theta_1, \ldots, \theta_n \rangle$. Any realization θ'_k of θ_k makes a crossing with $p \triangleright q$ that, in some sense, has $i-1$ traces between it and P and $n-i$ traces between it and Q. Suppose this crossing occurs at the point x. If θ'_k is to be part of a proper routing, x must be separated from both p and q as follows:

$$\|x - p\| \geq \mathit{width}(P)/2 + \mathit{width}(\theta_k)/2 + \sum_{i=1}^{k-1} \mathit{width}(\theta_i); \tag{1-1}$$

$$\|x - q\| \geq \mathit{width}(Q)/2 + \mathit{width}(\theta_k)/2 + \sum_{i=k+1}^{n} \mathit{width}(\theta_i). \tag{1-2}$$

The set of points x on $\overline{pq}$ satisfying these two inequalities is a **doorway** for the trace θ_i. It is empty if and only if the cut $\overline{pq}$ is unsafe. If the doorway $\overline{xy}$ of $\overline{pq}$ is not empty, the segments $\overline{px}$ and $\overline{qy}$ of its complement $\overline{pq} - \overline{xy}$ are called **struts** for θ_k.

Roughly speaking, we route each trace by finding the shortest route that passes through all of its doorways. To make this process finite, we consider only the doorways in certain special cuts.

Here we require that the wiring norm be piecewise linear. I assume that the routing algorithm is given the wiring norm in the form of its **unit polygon**, the set of vectors of norm 1. The unit polygon defines certain **diagonal slopes**, which are the slopes of the lines through the origin that contain vertices of the unit polygon. A cut is called **diagonal** if its slope is diagonal and one of its endpoints is a feature endpoint. The routing algorithm considers only the doorways in diagonal cuts.

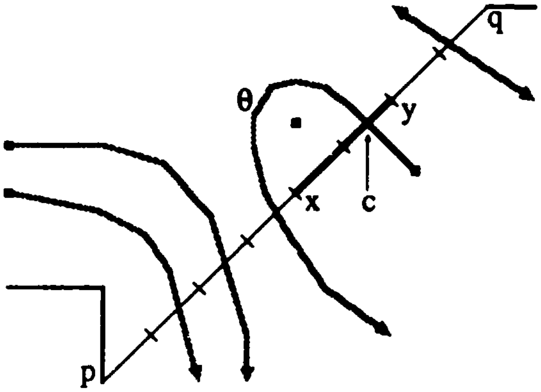


Figure 1d-1. *The doorway for a crossing of a diagonal cut.* All the traces (grey paths) have unit width, and the t.ck marks divide $\overline{pq}$ into segments of unit length. The line segment $\overline{xy}$ is the doorway for the crossing c between the cut $\overline{pq}$ and the trace θ.

(Hence its time and space complexity depend linearly on the number of vertices in the unit polygon.) In other words, the algorithm finds for each trace the shortest route that passes through its diagonal doorways.

To compute this route directly is difficult, so we do not consider all the doorways at once. Instead we consider only one diagonal slope at a time. The diagonal cuts of that slope split the routing region into trapezoidal strips; the rubber band for a trace passes through the strips in a particular order, and hence it has a particular sequence of doorways. These doorways form a corridor as shown in Figure 1d-2. The shortest path through this corridor is called a **partial realization** of the trace, though like rubber bands, it may touch features other than its terminals. The remarkable fact is that one can merge the partial realizations of a trace, one for each diagonal slope, to form the optimal, or *ideal*, realization of that trace. The complete routing algorithm is summarized below.

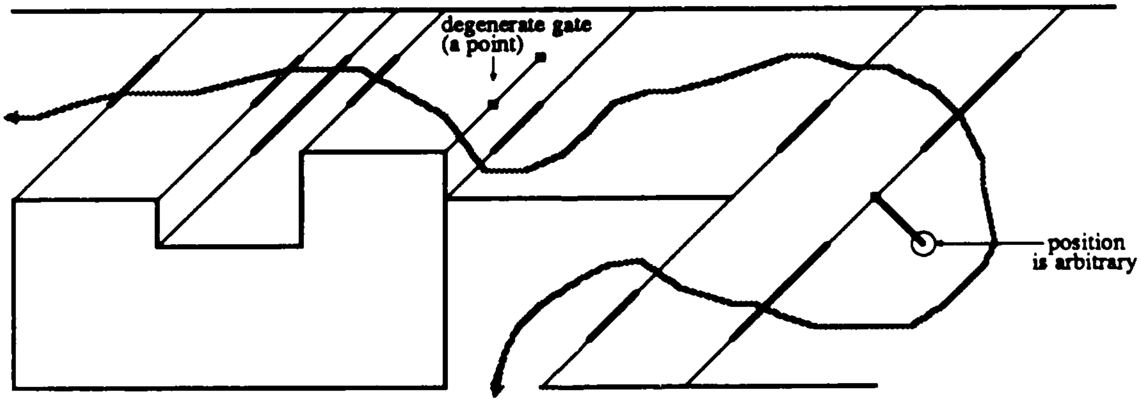


Figure 1d-2. *The corridor formed by a sequence of doorways for a trace.* Light lines represent diagonal cuts, and medium lines are features. Where consecutive doorways are collinear, an extra doorway is added to preserve the applicability of Algorithm W. A doorway may consist only of a single point, but it still contributes both left and right vertices to the corridor.

Algorithm R. (Produces a detailed routing of a routable sketch.)
Input: the RBE of a routable sketch S; the wiring norm's unit polygon.
Output: the ideal realization of each trace in S.
Local variables: array of partial realizations P.
Subroutines: Algorithm W is used in line 6.

```
1. foreach diagonal slope $s$ do
2.    Scan over the RBE with a line of slope $s$, producing doorways for all traces;
3.    foreach trace $\theta$ do
4.       $\rho \leftarrow$ rubber band of $\theta$;
5.       Sort the doorways of slope $s$ for $\rho$, producing a corridor;
6.       $P[\theta, s] \leftarrow$ shortest path through this corridor;
7. foreach trace $\theta$ do
8.    Merge the paths $P[\theta, s]$ to form the realization of $\theta$.
```

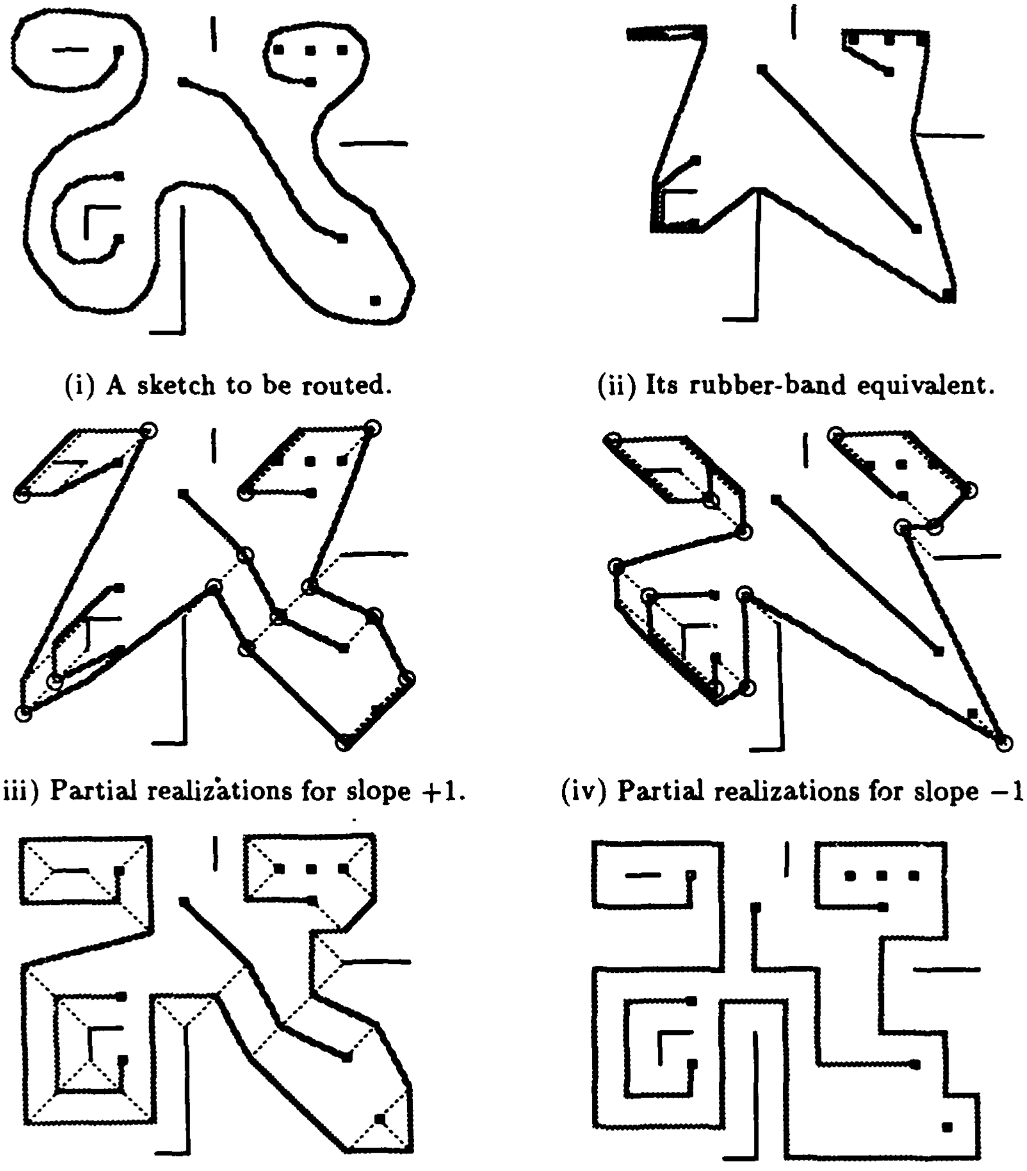
Constructing doorways and partial realizations

For each diagonal slope s, Algorithm R finds the doorways of slope s by scanning over the RBE with a line of slope s. The scanning technique is very similar to that used by Algorithm T, so I discuss it only briefly. We maintain the features and cables that cross the scan line in a pair of height-balanced trees. A feature enters or leaves the scan line when the scan line intersects one of its endpoints, say p. When this occurs we find the diagonal cuts of slope s incident on p by searching the feature tree for closest features not containing p. Having found a diagonal cut $p \triangleright q$, we determine the content of $p \triangleright q$ as explained in Section 1B. To do so we must know the sequence of cables that cross $p \triangleright q$ strictly between p and q; a search of the cable tree will provide this information quickly. Finally, from the content of $p \triangleright q$ we can construct the doorways for these traces in linear time. (The doorways are defined by equations (1–1) and (1–2).)

Next Algorithm R combines the doorways of each rubber band into a corridor for the corresponding trace. No sorting is actually required. Consider a trace θ with rubber band ρ. Each crossing of the diagonal cut $\overline{pq}$ by ρ can be associated with a particular strand of ρ. Hence every doorway for θ is associated with a point on a strand of ρ. By placing the doorways for each strand in a simple queue, the queues for ρ can be concatenated to yield the correct ordering of doorways for θ, and thus form a corridor. The shortest path through this corridor, which Algorithm W produces in linear time, is the partial realization of θ for the diagonal slope s.

Merging the partial realizations

In its final phase, Algorithm R combines the partial realizations of each trace to form the output traces. Let us define a **joint** of a piecewise linear path to be a



(i) A sketch to be routed.

(ii) Its rubber-band equivalent.

(iii) Partial realizations for slope +1.

(iv) Partial realizations for slope −1.

(v) The ideal realizations.

(vi) Realizations for a grid model.

Figure 1d-3. *The major steps in the routing of a sketch.* The wiring model here is rectilinear; its diagonal slopes are +1 and −1. All features and traces have unit width. Dark segments and points are features; grey lines are traces and their partial and ideal realizations; dashed segments are struts; and circles mark vertices of partial realizations that appear in the ideal realizations. Part (vi) shows that the ideal realizations can be altered so that they run in a grid. Algorithm R does not implement this process, but I discuss it in Chapter 10.

point where two segments of the path meet. The desired realization of a trace θ is a piecewise linear path whose joints are chosen from among the joints of the partial realizations of θ. Let σ denote the partial realization of θ for the diagonal slope s, and let ξ denote the ideal realization of θ.

There is a simple geometric procedure for determining whether a joint of σ is **retained** as a joint of ξ. Let $\overline{ax}$ and $\overline{xc}$ be consecutive segments of σ, with x the joint between them; then x is one endpoint of a doorway $\overline{xy}$ in a cut $\overline{pq}$. We say that σ **turns toward** p at x if p is not exterior to the angle $\angle axc$. The path σ turns toward either p or q at x, but not both. Assume σ turns toward p. Then x is retained if and only if the segments $\overline{ax}$ and $\overline{xc}$ touch the polygon $\{ z : \|z - q\| = \|x - q\| \}$ at x alone. To check this condition, it suffices to compare the slopes of $\overline{ax}$ and $\overline{xc}$ to the slopes of certain segments in the unit polygon of the wiring norm.

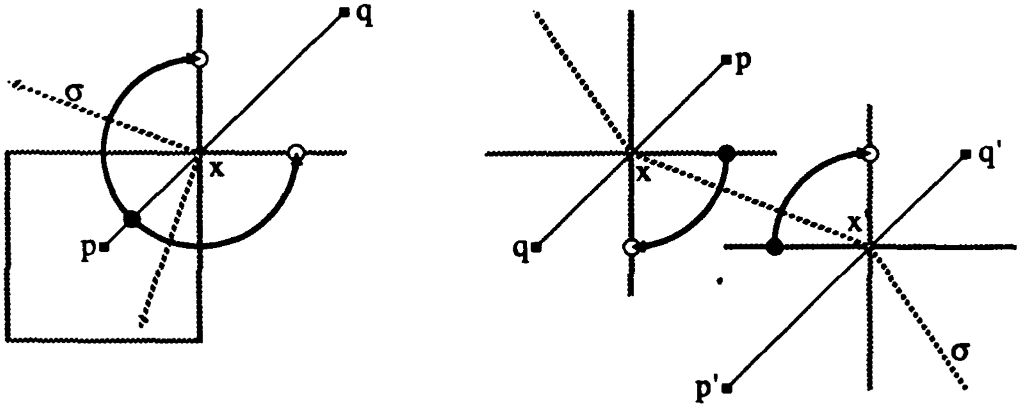


Figure 1d-4. *Evaluating the joints of a partial realization.* Here the partial realization σ has a joint x on the diagonal cut $\overline{pq}$, and σ turns toward p at x. We associate with x the two polygons $\{ z : \|z - p\| = \|x - p\| \}$ and $\{ z : \|z - q\| = \|x - q\| \}$, shown here in grey. Part (i) shows the range of angles that σ may make at x if x is to appear as a joint in the full realization. Similarly, part (ii) shows the range of angles that σ may make at consecutive joints x and x' if the segment $\overline{xx'}$ is to appear as a segment of the full realization.

It remains to find the correct ordering of the joints of ξ. Three simple rules govern this process. First, the joints of ξ that come from a partial realization σ have the same order and **orientation** in ξ as in σ. By the orientation of a joint I mean whether the path turns to the left or the right at the joint. Second, the joint of ξ that follows a joint x of σ is either another joint of σ, or else it comes from a partial realization σ' chosen as follows. If ξ turns left at x, then σ' corresponds to the next diagonal slope counterclockwise from s. Otherwise, if σ turns right at x, then σ' corresponds to the next diagonal slope clockwise from s. The third rule determines when two consecutive joints of σ are consecutive in ξ. Let x and x' denote these two joints, and let p and p' denote the corresponding feature endpoints toward which σ turns. The joints x and x' are consecutive in ξ if and only if the line segment between them intersects the polygons $\{\, z : \|z - p\| = \|x - p\| \,\}$ and $\{\, z' : \|z' - p'\| = \|x' - p'\| \,\}$ on their boundaries only. Again, this can be checked by comparing the slope of $\overline{xx'}$ to the slopes of certain segments in the unit polygon of the wiring norm.

An extension of the third rule allows the merging process to start and finish. Let t be the first (or last) terminal of θ. The first (or last) joint x of ξ has the property that the line segment $\overline{tx}$ intersects the polygon $\{\, z : \|z - p\| = \|x - p\| \,\}$ only on its boundary, where p is the feature endpoint toward which σ turns at x. Together these rules determine a unique piecewise linear path ξ. It can be produced in linear time from the partial realizations of θ, provided that the input sketch is routable.

Attempting to route an unroutable sketch

If the input to Algorithm R is the RBE of an unroutable sketch, then one of two things can happen. One possibility is that the process of merging partial realizations gets stuck: either it reaches a point where none of the available joints can be added, or it reaches the final terminal of the trace without having used all the available joints. The other possibility is that the merge completes successfully, but that the traces it has produced form an improper sketch. I conjecture that the latter possibility never arises, so that Algorithm R can always determine whether its input sketch is routable. If this conjecture proves true, then one need not apply Algorithm R to the input of Algorithm T to test for routability. One advantage of Algorithm T, however, is that it identifies the unsafe cuts that make the sketch unroutable. Algorithm R does not have this ability, and it can consume far more space than Algorithm T.

Complexity analysis

Algorithm R uses at most $O(|F|\,|T|)$ space to route a sketch (F,T). We mentioned in Section 1B that the detailed RBE is no larger than this. The number of doorways generated by phase one is also $O(|F|\,|T|)$, because each wire segment in the original sketch can cause at most one crossing of each diagonal cut. The output sketch fits in the same amount of space because its wire segments have endpoints on distinct doorways. On the other hand, sketches exist whose only detailed routings occupy $O(|F|\,|T|)$ space, so the space bound of Algorithm R is asymptotically optimal.

All the operations performed by Algorithm R take time linear in the size of its data structures, except the sorting that precedes the scanning operations, which requires logarithmic time per object. The time taken by Algorithm R is therefore at most $O(|F|\,|T|\log|S|)$. In practice, the number of crossings between diagonal cuts and wires should be much less than $|F|\,|T|$, and the algorithm should correspondingly faster. I have no experimental data to this effect, however.

1E. Efficiency Concerns

Seen from a theoretical standpoint, the algorithms for sketch routing and routability testing are quite efficient. Their worst-case running times are similar and seemingly close to optimal: $\Theta(n^2 \log n)$ on input of size n. In particular, the resource bounds of Algorithm R are optimal to within logarithmic factors on some inputs; there exist sketches of size n, like that in Figure 1e-1, whose only proper realizations have size $\Omega(n^2)$. From a practical standpoint, however, the sketch algorithms do not seem as good. Most programs that operate on integrated circuit designs have empirical running times close to linear, or at worst $O(n^{3/2})$ on input of size n. Since VLSI circuits are so huge, slower algorithms cannot be tolerated except when applied to small cells within a larger design.

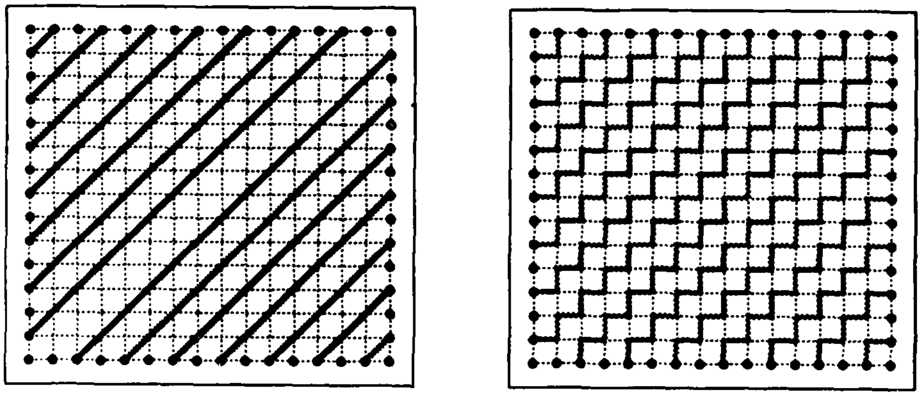


Figure 1e-1. *A small sketch whose proper realization is large.* If the distance between adjacent dotted lines is 1 unit and the unit polygon is square, then the only proper realization of the sketch on the left is the sketch on the right.

This section suggests two ways of speeding up routing and testing routability of sketches. One approach concerns worst-case performance. Algorithm R can be modified, without changing its underlying strategy, to eliminate the logarithmic factor in its time bound. Probably the running time of Algorithm T can also be improved to $O(n^2)$, at the cost of using $\Theta(n^2)$ space in every instance. The other approach concerns average-case performance. It begins with an investigation of the expected performance of Algorithms T and R on practical circuits, and then explores three methods for speeding up the performance bottleneck, which turns out to be Algorithm T. Two are described here; Section 1F is devoted to the third. All three methods involve paring down the number of critical cuts whose congestion needs to be computed. The result is a set of algorithms for the sketch routability problem whose average-case performance ranges from $\Theta(n \log n)$ to $\Theta(n^{3/2} \log n)$, depending upon assumptions concerning the placement of traces and features in a typical sketch. I should emphasize that these results are not based on any experimental evidence; I have not implemented any sketch algorithms. Instead I derive estimates of running time and space usage from models of the distribution of features and traces in the sketches input to Algorithms T and R.

Eliminating logarithmic factors

Recently I realized that the rubber-band equivalent is not the best data structure for Algorithm R to use in computing doorways. A faster method is to compute for each diagonal slope s a realization (not proper) whose traces cross the diagonal cuts of slope s as seldom as possible. Section 9B explains how to compute in time and space $O(|F|\,|T|)$ a structure called a *reduced intersection graph* that represents the necessary crossings of those traces and cuts. The content of a diagonal cut of slope s can be read off directly from the reduced intersection graph, as can the sequence of such cuts that each trace passes through. As a result the corridors for partial realizations can be computed in time $O(|F|\,|T|)$ per diagonal slope. Since the only logarithmic factors in Algorithm R came from constructing and scanning over the RBE, the result is an algorithm for sketch routing that runs in time $O(|F|\,|T|)$ plus $O(|F| \log |F|)$ to scan for diagonal cuts. In fact, its running time is essentially proportional to the number of crossings between traces in the input sketch and diagonal cuts. The only reason not to adopt this approach is that my sketch compaction algorithm, currently the only client of Algorithm R, applies Algorithm R to a rubber-band equivalent rather than a sketch.

A similar improvement may be possible in Algorithm T. Leo Guibas [15] has suggested that the scanning in Algorithm T can be replaced by a topological sweep [11], reducing the worst-case running time from $O(n^2 \log n)$ to $O(n^2)$. To take advantage of this speedup, and to obtain a similar speedup in the construction of the condensed RBE, that structure must be represented in the form of an adjacency matrix. Hence $\Theta(n^2)$ space is required in every instance, as opposed to $\Theta(n)$ space for Algorithm T as it stands. Consequently this improvement is of academic interest only.

The proper measure of input size

The sketch algorithms so far described—for constructing the RBE, testing routability, and routing—all run in essentially quadratic time, but this running time arises from different causes in each case. When constructing the RBE, the number of crossings between trace segments and doors determines the running time to within a logarithmic factor. When testing routability, the time complexity is determined by the number of pairs of features in the condensed RBE, again with an added logarithmic factor. And when routing a sketch, the relevant quantities are the number of strands in the RBE and the number of crossings between trace segments and diagonal cuts. Of these quantities, only the number of pairs of features is generally $\Theta(n^2)$. Likewise, though the space usage of Algorithm R is $\Theta(n^2)$ in the worst case, it is actually proportional to the number of crossings between traces and certain cuts (doors and diagonal cuts).

I argue that a sketch algorithm whose resource usage is nearly proportional to the number of crossings between traces and $O(n)$ cuts is really quite efficient. Whether the expected number of such crossings is close to linear in n depends on one's source of sketches. But in any case, that quantity is a more reasonable measure of sketch complexity than n, the number of feature and trace segments in the sketch. The reason is that one can encode quite complicated sketches with just a few segments. Each trace segment in the input sketch can, in principle, span the width of the sketch. One would prefer a measure of sketch size that accounted for the lengths of trace segments. Probably no such measure is convenient, but if one adopts this viewpoint, the complexity of Algorithm R, in particular, seems much smaller. So the only algorithm that could really stand improvement is Algorithm T. The bottleneck is the repeated scanning around feature endpoints for critical cuts, which takes $O(n \log n)$ time per feature whether such critical cuts are found or not.

Typical sketches

What properties of practical sketches can we exploit to speed up Algorithm T? I submit that there are at least three: *density*, *uniformity*, and *locality*. By density I mean that the components in typical circuit layouts are tightly packed. Depending on how the layout components are represented in sketches, the only features visible from a given feature may be a few of its nearest neighbors. (If almost all features are points, then visibility is essentially unlimited. But if many features are line segments, then expected visibility is bounded independently of sketch size. This fact is independent of density, but the constant—the expected number of features visible from a given feature—does depend on density.) In this case the expected number of critical cuts is $\Theta(n)$. The second principle, uniformity, says that the elements of a sketch are distributed nearly uniformly in a rectangular region of small aspect ratio. Applied to the rubber-band equivalent, it implies that the average number of cables crossed by a critical cut is $O(1)$ if visibility is restricted, and $O(\sqrt{n})$ if visibility is unrestricted. Finally, locality suggests that local constraints almost always dominate over nonlocal ones. In other words, it is highly unlikely that a long nonempty critical cut is unsafe without some shorter nonempty critical cut being unsafe also. None of these principles can be justified formally, but I think that programmers of circuit design systems will agree that they are reasonable assumptions.

Checking critical cuts: two approaches

The locality principle has immediate application to routability testing. Rather than scanning the entire sketch from each feature endpoint, one could scan only part of the sketch each time. For example, one might first divide the components of the RBE into bins, each bin corresponding to a square region of the sketch. For each feature endpoint, one could then include only the components in its bin and the adjacent bins in the scanning operation. This technique should be fast, but it has the drawback of relying on the locality principle for correctness, not just performance. If an unsafe, nonempty, critical cut is found, the sketch is proven to be unroutable. But if no such cut is found, the sketch is not necessarily routable. Finding a good tradeoff between speed and risk of error would probably require extensive experimentation.

A less risky approach to routability testing relies instead on the assumption of density. Rather than locating the critical cuts by scanning, we obtain them from the **visibility graph** of the sketch. The visibility graph (V, E) of a sketch (F, T) is a graph whose vertices are the endpoints of features in F and whose edges are the line segments in F and all straight cuts between endpoints of features in F. The edges emanating from each feature endpoint are sorted in clockwise order. One can compute the visibility in time $\Theta(|E| + |F| \log |F|)$ and space $\Theta(|E|)$ by the methods of [14]. The running time averages $\Theta(|F| \log |F|)$ if our sketch is dense, meaning that the expected number of cuts between feature endpoints is $\Theta(|F|)$, and is $\Theta(|F|^2)$ in the worst case. Given the visibility graph (V, E) of a sketch, the critical cuts can be enumerated in time $\Theta(|E|)$. For each feature endpoint p, one can list the portions of features visible from that endpoint, and check which of those portions contain the closest points to p on their respective features.

Having identified the critical cuts, one must compute their congestions without scanning. The simplest way to do so, based on what we already know, is to make separate queries to the condensed RBE for each critical cut. The condensed rubber-band equivalent of a sketch is like an embedded planar multigraph. We may consider it one since although some of its arcs overlap, they are circularly ordered at the nodes they connect. Some of the faces of this graph are polygonal, and some are degenerate (where two edges connect the same feature endpoints). In $O(|F|\log|F|)$ time we can add edges so as to triangulate the polygonal faces, keeping the size of the whole graph linear in $|F|$. Now we compute the dual graph: the graph whose nodes are the faces of the original graph and whose edges represent adjacency across the original edges. This computation takes linear time.

Every cut now corresponds to a path in the dual graph whose length is the number of cables crossed over by the cut. One can find this path in time proportional to its length, simply by walking through the dual graph. The congestion of the cut $\overline{pq}$ is the sum of the widths of the cables that $\overline{pq}$ crosses over, plus the widths of the crossing sequences of $\vec{pq}$ and $\vec{qp}$. (As usual, when a cable lies within $\overline{pq}$, it may or may not contribute to $cong(\overline{pq})$, depending on which of the three possible cables from p to q it is.) The crossing sequence terms are provided by the condensed RBE at a cost of $O(\log n)$ time per cut. We conclude that after $O(n\log n)$ preprocessing operations on the condensed RBE, the congestion of a cut can be computed in time $O(\log n)$ plus $O(1)$ per cable it crosses over.

Using both data structures—the visibility graph and dual of the condensed RBE—we obtain an algorithm for testing sketch routability whose performance is potentially far superior to that of Algorithm T. Under the most optimistic assumptions of density and uniformity, the expected running time is $\Theta(n\log n)$. In the very worst case, $\Theta(n^3)$ time might be consumed. One drawback to this approach is the complexity of implementing the algorithm that constructs the visibility graph.

1F. Faster Routability Testing

This section describes a very powerful technique for speeding up routability testing in piecewise wiring norms, the kind we use. It results in a routability testing algorithm that runs in time $O(n^{3/2}\log n)$ on typical sketches of size n, without needing any more than linear space.

The key to routability testing is finding a small set of decisive cuts. So far we have considered methods for identifying and checking the *critical* cuts in a sketch. Critical cuts are decisive but difficult to enumerate, since every pair of features can potentially generate a critical cut. To determine which of the minimum-length paths between features are actually critical cuts, one must either consider all pairs of features (as Algorithm T does by scanning) or construct something like a visibility graph. By exploiting a property of cuts under piecewise linear norms, we can eliminate many pairs of features from consideration, whether or not they generate a critical cut. This property, called *shadowing*, was discovered by Cole and Siegel [6] and independently by me. Some line segments in the sketch are *shadowed* by other features, and even if they are critical cuts, they need not be checked. If the arrangement of features in the sketch is close to uniform, then most cuts are shadowed, and one can quickly generate a decisive set of unshadowed cuts.

Definition of shadowing

The principle of shadowing is that no cut $\overline{pq}$ need be checked if there is a point r on a feature such that

$$\|p-q\| = \|p-r\| + \|q-r\| . \tag{1-3}$$

If the point p is considered fixed, the point r casts a **shadow** consisting of all points q which, together with p and r, satisfy (1–3). We say that the cut $\overline{pq}$ is **shadowed** (by r). Typical shadows for the rectilinear (L^∞) wiring norm are pictured in Figure 1f-1. If the norm $\|\cdot\|$ were the euclidean norm, this shadow would be nothing more than the ray starting at r and pointing away from p. But since the wiring norm is piecewise linear, shadows can have substantial size. More to the point, if the features in a sketch are evenly distributed, then the number of unshadowed features, as seen from a given feature endpoint, is likely to be small: $O(\log|F|)$ on the average. Later in this section I justify this bound and explain why shadowed cuts may be ignored.

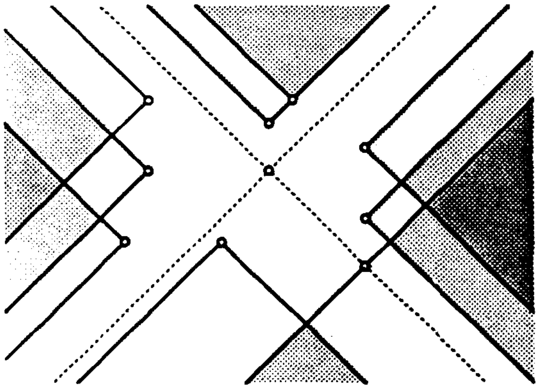


Figure 1f-1. *Shadows in the rectilinear norm.* With respect to the central point, each of the other points casts a shadow, shown as a shaded region. Shadows include their frontiers. Darker shades represent multiple overlapping shadows. The dashed lines are the lines of diagonal slope passing through the central point. Points on these lines are shadowed only by other such points.

We use shadowing to find a small decisive set of cuts for a sketch. This set contains the diagonal cuts in the sketch, of which there are $O(|F|)$, and the unshadowed cuts between feature endpoints, of which there are typically $O(|F|\log|F|)$. The expected time needed to find these cuts is also $O(|F|\log|F|)$. One can compute the congestion of the decisive cuts from the dual of the condensed RBE, as described in Section 1E, at an average cost of $O(\sqrt{|F|})$ time per cut. If the cuts are checked as they are produced, none of our data structures grows larger than $O(|F|)$. The result is an algorithm for sketch routability that consumes only linear space and $O(|F|^{3/2}\log|F|)$ time, plus that needed to construct the condensed RBE, for typical sketches.

Scanning for unshadowed cuts

Simple scanning algorithms suffice for finding the decisive cut set. Diagonal cuts, in particular, are easy to find by scanning with lines of diagonal slope as Algorithm R does. I now present an algorithm that enumerates the other desired cuts. Shadowing works best when there are only two diagonal slopes, as when the wiring norm is rectilinear. In this case, scanning for unshadowed cuts takes time $O(|F|\log|F|)$ plus $O(1)$ per cut found. For simplicity I illustrate the algorithm using the taxicab (L^1) norm, defined by $\|(x,y)\| = |x| + |y|$, which is the rectilinear norm rotated through $\pi/4$ radians and rescaled by $\sqrt{2}$. In the taxicab norm the points that can shadow a cut $\overline{pq}$ are the points in the rectangle whose sides are aligned with the axes and which has p and q at opposite corners. I also simplify matters by assuming that all features are points. It matters little if the algorithm outputs a line segment that is not really a cut, because the fact that it is not a cut will be discovered when trying to compute its congestion.

We compute all the unshadowed cuts between feature endpoints by scanning over the sketch from left to right with a vertical line. Actually, we skip some unshadowed cuts that are diagonal and produce some cuts that are just on the boundary of being shadowed, but these discrepancies cause no problems. The scan considers only feature endpoints. At all times during the scan, we maintain a data structure that contains every feature endpoint lying left of the scan line, except that where two or more feature endpoints have the same y-coordinate, only the rightmost is kept. These feature endpoints are kept sorted by y-coordinate, presumably in some height-balanced tree to enable fast insertion. Each point in the structure also has two links to other points in the structure, an **upward** link and a **downward** link. The upward link of p points to the feature endpoint above it and strictly to its right that is closest to p in y-coordinate. If no such point exists, then the upward link is *nil*. The downward link of p is similar, but points to the closest feature endpoint below it and strictly to its right. See Figure 1f-2(a).

Adding a new feature endpoint to the structure is simple. Figure 1f-2(b) illustrates the process. Call the new endpoint q. One first finds the feature endpoints left of q that lie just above and below q in y-coordinate. Denote them by p^+ and p^-, respectively. They are identical if q has the same y-coordinate as a point already considered. Next one finds the unshadowed cuts incident on q from the left, while at the same time updating the up and down links of the existing points. Beginning at p^+, follow the upward links until reaching *nil* or a point on the scan line. All the points in this chain have unshadowed cuts to q, and their downward links must be

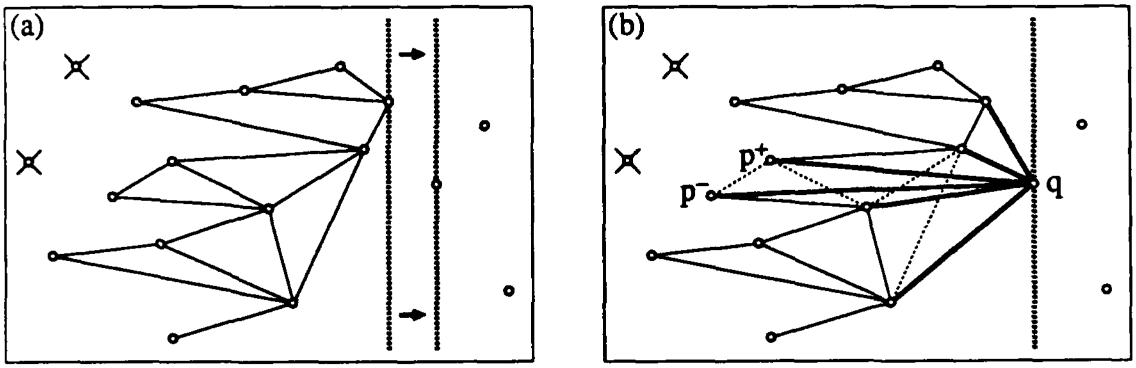


Figure 1f-2. *Scanning for unshadowed cuts.* As the scan line (dark vertical line) moves to the right, the network of upward and downward links (light lines) is updated. All links point to the right; null pointers are not shown. Crossed-out points have been superseded by points farther to the right having the same y-coordinates. In (b), the pointers shown as dotted lines are being replaced by the darker links, which also represent the unshadowed cuts incident on q from the left.

modified to point to q. Next, starting at p^-, follow the down links until reaching *nil* or a point on the scan line. All the points in this chain have unshadowed cuts to q, and their upward links should now connect them to q. If $p^+ = p^-$, one deletes this point. Finally one sets the upward and downward links of q to *nil*.

The correctness and complexity analysis of this method are straightforward. Processing one feature endpoint takes time $O(\log |F|)$ per feature endpoint plus $O(1)$ time per cut produced. Applied to each feature endpoint in turn, it produces all the unshadowed cuts between them except those that are vertical. In the taxicab norm, vertical cuts between feature endpoints are diagonal cuts. Because the diagonal cuts are gathered separately, there is no harm in ignoring them here. Similarly, the horizontal cuts generated in the scan for unshadowed cuts may be dropped to avoid duplication.

More complicated wiring norms

No fundamental changes are needed in the scanning algorithm if the unit polygon of the wiring norm is a parallelogram and not a square. The unit polygon is a parallelogram if and only if the wiring norm has exactly two diagonal slopes. One simply redefines one diagonal slope to be "vertical" and the other to be "horizontal", and reinterprets the terms 'above', 'below', 'left', and 'right' accordingly. Equivalently, one may rotate and skew the sketch and its wiring norm so that one diagonal slope actually *is* vertical and the other is horizontal, and apply the inverse transformation to each cut generated.

Scanning for unshadowed cuts is somewhat more complicated when the unit polygon of the wiring norm has more than four sides. In this case several scans are needed to produce all the cuts. Each scan produces the unshadowed cuts whose slopes lie in a certain range. For each diagonal slope s, we need a scan that generates the unshadowed cuts whose slopes lie between s and the diagonal slope t immediately clockwise from s. In this scan we pretend that s and t are the only diagonal slopes, and throw away the generated cuts whose slopes do not lie clockwise between s and t. (Which cuts are shadowed is independent of all properties of the wiring norm except the diagonal slopes.) The remaining unshadowed cuts are, in fact, unshadowed in the original wiring norm. Since some of the generated cuts have to be thrown away, we can no longer claim that the scanning algorithm runs in time $O(|F|\log|F|)$ plus $O(1)$ per unshadowed cut. Nevertheless, the average number of cuts thrown away, as well as the average number retained, is only $O(|F|\log|F|)$ per diagonal slope. We now justify this figure.

The number of unshadowed cuts

The average-case time complexity of the new routability testing method depends foremost on the number of unshadowed cuts. More accurately, it depends on the number of line segments output by the scanning procedure, which is approximately the number of unshadowed cuts that would exist if each feature endpoint were a feature unto itself, i.e., if all features were points. We now estimate this quantity; it turns out to be $O(n \log n)$ where $n = \Theta(|F|)$ is the number of feature endpoints. As a model of the distribution of feature endpoints, we assume that n points are independently and uniformly distributed in the unit square $I \times I$. The size of the square is irrelevant to the present analysis. We wish to estimate the expected number of pairs (p, q) of these points for which the cut $\overline{pq}$ unshadowed. Since expectation is additive, regardless of independence, it suffices to determine the chance that a particular cut $\overline{pq}$ is unshadowed, and multiply this chance by $\binom{n}{2}$.

An approximate analysis shows that the probability of a cut being unshadowed is $O(\ln(n)/n)$. Let p and q be two of the randomly placed points, and let $\Box pq$ denote the rectangular region with diagonal $\overline{pq}$ whose sides are aligned with the axes. Define a random variable A whose value is the area of $\Box pq$. The cut $\overline{pq}$ is output by the scanning procedure only if the inside of $\Box pq$ contains none of the n points except p and q. This event has probability $(1 - A)^{n-2}$. Define the random variables X and Y to be the horizontal and vertical separations, respectively, of p and q. Almost all the contribution to the chance that $\Box pq$ is empty comes from small X and Y. We are willing to ignore constant factors, so there is no harm in pretending that X and Y are uniformly distributed in $[0, 1]$. If this were true, then for $a \in [0,1]$ we would have

$$\begin{aligned}\Pr[A \le a] &= \iint_{xy \le a} 1\,dy\,dx \\ &= \int_0^a \int_0^1 1\,dy\,dx + \int_a^1 \int_0^{a/x} 1\,dy\,dx \\ &= a + \int_a^1 a/x\,dx = a(1 - \ln a).\end{aligned}$$

Differentiating with respect to a gives $-\ln a$, so $a \mapsto -\ln a$ is a good estimate of the density function for A. Hence the probability that $\overline{pq}$ is unshadowed is on the order of

$$\int_0^1 (1-a)^{n-2}(-\ln a)\,da = -\int_0^1 u^{n-2}\ln(1-u)\,du.$$

Now we integrate by parts, choosing the antiderivative $(u^{n-1}-1)/(n-1)$ for u^{n-2}, and thus obtain

$$-\int_0^1 u^{n-2}\ln(1-u) = -\left[\frac{u^{n-1}-1}{n-1}\ln(1-u)\right]_0^1 + \int_0^1 \frac{u^{n-1}-1}{(n-1)} \cdot \frac{-1}{1-u}\,du.$$

The bracketed term vanishes, and we are left with the integral

$$\begin{aligned}\frac{1}{n-1}\int_0^1 \frac{u^{n-1}-1}{u-1}\,du &= \frac{1}{n-1}\int_0^1 (1+u+u^2+\cdots+u^{n-2})\,du \\ &= \frac{1}{n-1}\sum_{k=1}^{n-1}\frac{1}{k} \approx \frac{\ln n}{n}.\end{aligned}$$

The expected number of unshadowed cuts among the n points is therefore $\binom{n}{2}$ times $O(\ln(n)/n)$, which is $O(n \log n)$.

This analysis changes only quantitatively, not qualitatively, if the wiring norm is not the taxicab norm. Because of the way we break down a complicated wiring norm into wiring norms with two diagonal slopes each, it suffices to consider a wiring norm whose unit polygon is an arbitrary parallelogram. Then the points that can shadow a cut $\overline{pq}$ all lie in a parallelogram $\Diamond pq$ with p and q at opposite corners. This parallelogram takes the place of $\Box pq$. The distribution of the area of $\Diamond pq$ is asymptotic to that of $\Box pq$ as area approaches 0, and hence the chance that $\overline{pq}$ is unshadowed remains $O(\ln(n)/n)$. Therefore the expected number of cuts generated while scanning is $O(n \log n)$ per diagonal slope.

An explanation of shadowing

Why should the unshadowed cuts between feature endpoints and the diagonal cuts form a decisive set? The answer to this question has two parts. The first part notes that the set of all cuts between feature endpoints, together with the diagonal cuts, constitute a decisive set. For lack of a better word, let us call these cuts **pivotal**. Pivotal cuts are strongly related to critical cuts. Recall that a critical cut is a cut from a feature endpoint to the closest point on another feature as measured in the wiring norm, with ties broken using the euclidean norm. As you might guess, the method of tiebreaking is arbitrary. Instead one can use a tiebreaker that always picks a diagonal cut or a cut between feature endpoints. For if p is any point and Q is a feature not containing p, there is a point $q \in Q$ minimizing $\|q - p\|$ such that either q is an endpoint of Q or the slope of $\overline{pq}$ is diagonal. Hence the pivotal cuts form a decisive set for the same reason that critical cuts do. (See Proposition 8b.4.)

The second part of the answer explains why shadowed cuts need not be checked. Shadowing derives its power from the following lemma.

Lemma: *Let $\overline{pq}$, $\overline{pr}$, and $\overline{qr}$ be cuts in a sketch. Assume that r shadows $\overline{pq}$ and that the inside of the triangle $\triangle pqr$ is free of features. If $\overline{pq}$ is unsafe and nonempty, then either $\overline{pr}$ or $\overline{qr}$ is unsafe and nonempty.*

The idea behind the lemma is the following. Let P, Q, and R denote the islands containing p, q, and r, respectively. Then by equation (1–3) and the definition of capacity, we have

$$\text{capacity of } \overline{pq} = \text{capacity of } \overline{pr} + \text{capacity of } \overline{qr} + \text{width of } R. \qquad (1\text{–}4)$$

On the other hand, Figure 1f-3 suggests that

$$\text{congestion of } \overline{pq} \leq \text{congestion of } \overline{pr} + \text{congestion of } \overline{qr} + \text{width of } R, \qquad (1\text{–}5)$$

since the trace ρ is no wider than its terminal R. This inequality can be proven using the machinery of Section 4F (Proposition 4f.1) and Chapter 8. Subtracting the relation (1–5) from equation (1–4), we infer that the *margin of safety* of $\overline{pq}$, the difference between its capacity and congestion, is at most the sum of the margins of safety of $\overline{pr}$ and $\overline{qr}$. Hence if $\overline{pq}$ is unsafe—if its margin of safety is negative—then either $\overline{pr}$ or $\overline{qr}$ is unsafe. Moreover, if $\overline{pq}$ is also nonempty, one of $\overline{pr}$ and $\overline{qr}$ is unsafe and nonempty. I leave this last deduction as an exercise.

The lemma gives us a condition under which a shadowed cut $\overline{pq}$ need not be checked. Suppose the cut $\overline{pq}$ is shadowed by a point s. Let r be the closest point on a feature to $\overline{pq}$, excluding p and q, in the closed region bounded by the triangle $\triangle pqs$. Then $\overline{pr}$ and $\overline{qr}$ are cuts, the inside of $\triangle pqr$ is empty, and r shadows $\overline{pq}$.

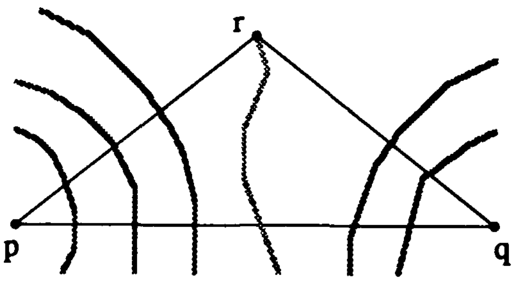


Figure 1f-3. *An inequality concerning congestion.* If the inside of Δpqr is free of features, then all traces (darkly shaded paths) that necessarily cross $\overline{pq}$ also necessarily cross either $\overline{pr}$ or $\overline{qr}$, with at most one exception: one trace (lightly shaded path) can have r as a terminal.

Hence the lemma applies to p, q, and r. It shows that $\overline{pq}$ cannot be unsafe and nonempty unless either $\overline{pr}$ or $\overline{qr}$ has the same properties. So we can avoid checking $\overline{pq}$ if we can determine that neither $\overline{pr}$ nor $\overline{qr}$ is both unsafe and nonempty.

Now we invoke a special property of pivotal cuts, which can be traced all the way to Lemma 7c.3. If any cut in the sketch is unsafe and nonempty, then the sketch has a pivotal cut that is also unsafe and nonempty, simply because the pivotal cuts are decisive. The special property is this: the unsafe and nonempty pivotal cut is *no longer* than the original cut, as measured by the wiring norm. To ensure that $\overline{pr}$ and $\overline{qr}$ are safe, it therefore suffices to check pivotal cuts that are shorter than $\overline{pr}$ and $\overline{qr}$, and are therefore shorter than $\overline{pq}$. In other words, we may remove $\overline{pq}$ from our decisive cut set, which consisted originally of the pivotal cuts. The same principle applies to all shadowed pivotal cuts, and hence the unshadowed pivotal cuts form a decisive set. This set contains precisely all the unshadowed cuts between feature endpoints and all the diagonal cuts, since diagonal cuts are never shadowed.

Chapter 2

Topological Preliminaries

Point-set topology and elementary homotopy theory form the basis for all the mathematical work in this thesis. Since point-set topology is more widely known, and too large a subject to be covered here, I assume familiarity with its basic concepts and the relationships among them. The reader should know the definition of the terms *basis*, *component*, *embedding*, *homeomorphism*, *path*, and *quotient space*, the concept of a *local* property, and what it means for a space to be *compact*, *connected*, *Hausdorff*, *metric*, *normal*, or *path-connected*. For those readers who wish to refresh their memories, I have provided definitions of these terms in the glossary. An excellent reference, both for point-set topology and for an introduction to homotopy theory, is the text by Munkres [38].

Unlike the other chapters, this chapter contains little or no original material; it merely encapsulates known results for future reference. As the nomenclature of topology is not entirely standardized, the first part of the chapter describes the terms and symbols I have adopted. *Everyone who wishes to study the mathematical parts of this thesis should read these definitions*, because Chapters 3 through 8 depend on them. The rest of the chapter reviews some elementary results from different branches of topology. Sections 2A and 2B introduce homotopy theory at an elementary level. I have provided proofs for the easier results to help the reader assimilate the definitions. Section 2C discusses some facts about plane curves that will be used from time to time. Lastly, Section 2D distills the results we will need concerning topological manifolds. The proofs in the final section rely on the machinery of homology theory, but no homology theory is used elsewhere in this thesis.

Topological spaces and maps

A **space** always means a topological space, and a **map** on topological spaces always means a continuous function. The following are standard topological spaces:

- the unit interval $I = [0, 1]$,
- the euclidean spaces R^n, for $n \geq 1$,

- the euclidean half-spaces $H^n = \{(x_1, \ldots, x_n) \in R^n : x_n \geq 0\}$, for $n \geq 1$,
- the spheres $S^n = \{x \in R^{n+1} : |x| = 1\}$, for $n \geq 0$.

In every case, the superscript denotes the topological dimension of the space itself, and not the dimension of the space in which it is embedded. I reserve the right to use each of the symbols R, H, and S without a superscript to mean something other than the spaces listed above. In particular, R^1 should be distinguished from R, which need not denote the real line. When a space such as $\{x\}$ contains only one element, I frequently omit the braces and write simply x.

A subspace $A \subseteq X$ is a **retract** of X if there is a map $r: X \to A$, called a **retraction**, such that $r(a) = a$ for all $a \in A$. The spaces I and R^1 are **absolute retracts** in the following sense. If I or R^1 is embedded in a normal space X as a closed subspace A, then A is a retract of X.

For every subspace A of a topological space X, the following subspaces of X are defined.

- Its **interior** $\mathit{Int}\, A$, the union of the open sets contained in A.
- Its **closure** $\mathit{Cl}\, A$, the intersection of the closed sets that contain A.
- Its **frontier**, or "topological boundary", which is $\mathit{Fr}\, A = \mathit{Cl}\, A - \mathit{Int}\, A$.

The term 'boundary' is reserved for use with manifolds.

I employ a few convenient devices for describing maps. If $E(t)$ is any expression, then '$t \mapsto E(t)$' denotes the function whose value at t is $E(t)$. The domain and range of this function should be inferred from context. If $F: X \times Y \to Z$ is a function with two arguments, then $F(x_0, \cdot): Y \to Z$ is the function $y \mapsto F(x_0, y)$, and $F(\cdot, y_0): X \to Z$ is the function $x \mapsto F(x, y_0)$. This "dot" notation generalizes to more complicated expressions. If $f: X \to Y$ and $U \subseteq X$, then $f|_U$ denotes the restriction of f to U. We write $f(U)$ for the set $\mathit{Im}\, f|_U = \{f(u) : u \in U\}$. The symbol id_X denotes the identity map on the space X.

Paths and their images

A path α is always a continuous function with domain I, and should not be confused with its image $\mathit{Im}\, \alpha$. When we speak of a path intersecting a set or another path, however, we are implicitly referring to the image of that path. The **endpoints** of a path α are the points $\alpha(0)$ and $\alpha(1)$, and are considered as an ordered pair. Thus α and β **have the same endpoints** if $\alpha(0) = \beta(0)$ and $\alpha(1) = \beta(1)$. I write $\alpha: A \rightsquigarrow B$ to mean that α is a path with $\alpha(0) \in A$ and $\alpha(1) \in B$. The **middle** of the path α is the set $\mathit{Mid}\, \alpha = \alpha\big((0,1)\big)$.

The distinction between paths and their images is reflected by the distinction between linear paths and line segments. In a space such as R^n that has a linear structure, the **linear path** $x \triangleright y$ is the path $t \mapsto (1-t)x + ty$, whereas the **line segment** $\overline{xy}$ is the set $\mathit{Im}(x \triangleright y)$. We say α is **piecewise linear** if there is a partition

$0 = t_0 < t_1 < \cdots < t_n = 1$ of I such that α is linear on each interval $[t_{i-1}, t_i]$. If this partition is minimal, so that α is not linear on any interval $[t_{i-2}, t_i]$, then we call the points $\alpha(t_i)$ the **vertices** of α. A piecewise linear, injective path is called **simple**. A **loop** is a path whose endpoints coincide; the loop α is **simple** if α is piecewise linear and $\alpha(s) = \alpha(t)$ implies $s = t$ or $\{s, t\} = \{0, 1\}$. A **polygon** is either a simple loop in R^2 or the image of such a loop, depending on context. A subset of R^2 is **polygonal** if it comprises the inside of a polygon together with some or all of its frontier.

I now define three important operations on paths. If α is a path in X, and $a, b \in I$, then the path obtained by varying the argument of α from a to b is the path $\alpha_{a:b}: I \to X$ given by

$$\alpha_{a:b}(t) = \alpha\bigl((1-t)a + tb\bigr).$$

We call $\alpha_{a:b}$ a **subpath** of α. If α and β are paths in X satisfying $\alpha(1) = \beta(0)$, then their **concatenation** is the path $\alpha \star \beta: I \to X$ equaling

$$t \mapsto \begin{cases} \alpha(2t), & \text{if } t \le \frac{1}{2}; \\ \beta(2t-1), & \text{if } t \ge \frac{1}{2}. \end{cases}$$

Note that $(\alpha \star \beta)_{0:1/2} = \alpha$ and $(\alpha \star \beta)_{1/2:1} = \beta$. The **reverse** of a path α, denoted $\widehat{\alpha}$, is $\alpha_{1:0}$.

Given a way to measure the length of a linear path, I define the **arc length** of a path α to be the least upper bound of the lengths of piecewise linear approximations to α. (A piecewise linear path β approximates α if $\beta(t) = \alpha(t)$ for each vertex $\beta(t)$ of β.) If α is a path in R^2, the euclidean arc length of α is denoted $|\alpha|$. The arc length of α in an arbitrary norm $\|\cdot\|$ is denoted $\|\alpha\|$. One reason for using norms rather than arbitrary metrics is to make the arc length of a linear path equal the distance between its endpoints: in any norm $\|\cdot\|$ we have $\|p \triangleright q\| = \|p - q\|$. A path α of finite arc length is **canonical** if $|\alpha_{0:t}| = t \cdot |\alpha|$ for every $t \in (0, 1]$.

Geometric primitives

Because piecewise linear paths are central to this work, we need a few more definitions relating to them. Some of these definitions supersede less precise definitions given in Section 1D. Let α be a piecewise linear path. A **joint** of α is a point $s \in (0, 1)$ such that for no open interval (x, y) containing s is the subpath $\alpha_{x:y}$ linear. A **segment** of α is a subpath $\alpha_{s:t}$ with $s < t$ such that each of s and t is a joint of α or a point in $\{0, 1\}$. Now let $\alpha_{r:s}$ and $\alpha_{s:t}$ be consecutive segments of a piecewise linear path $\alpha: I \to R^2$. We say that α **turns at s** if neither $\alpha_{r:s}$ nor $\alpha_{s:t}$ is constant, and $\alpha(s)$ does not lie on the linear path $\alpha(r) \triangleright \alpha(t)$. If α turns at s, then the rays from $\alpha(s)$ through $\alpha(r)$ and $\alpha(t)$ form an angle of measure less than π (and perhaps of measure 0). The path σ **turns away from** a point $x \in R^2$ at s if x is exterior to this angle, and otherwise σ **turns toward** x at s.

Whenever a path has two "sides", it makes sense to talk about another path crossing *over* it. And at least in the neighborhood of any point on a nonconstant segment, every piecewise linear path does have two sides. We say that α **crosses over** β at a point $x \in I$ if there is an interval $[s,t]$ containing x such that $Im\, \alpha_{s:t} \subset \beta$ but the paths $\alpha_{0:s}$ and $\alpha_{1:t}$ approach β from opposite sides.

2A. Homotopies and the Fundamental Group

The notion of a rough routing for a wire is rooted in the mathematical idea of *path homotopy*. Hence in the of study planar wiring problems involving rough routings, we look first at the homotopy theory of paths. This section defines the appropriate notions of homotopy for paths and general maps, gives a precise definition of *simple connectivity*, and provides several methods for proving that a space is simply connected.

Path homotopy

Roughly speaking, two paths are *path homotopic* if one can be continuously deformed into the other without moving its endpoints. One can make this notion precise by expressing the continuous deformation as a continuous function.

Definition 2a.1. Two paths $\alpha, \beta: I \to Y$ are **path homotopic**, denoted $\alpha \simeq_P \beta$, if there is a map $F: I \times I \to Y$ such that $F(\cdot,0) = \alpha$, $F(\cdot,1) = \beta$, and the maps $F(0,\cdot)$ and $F(1,\cdot)$ are constant. The map F is called a **path homotopy** between α and β.

A path homotopy F defines a family of paths $\{F(\cdot,t) : t \in I\}$ with the same endpoints. As t varies from 0 to 1, the path $F(\cdot,t)$ varies in a continuous manner. A good example of a path homotopy is given by the following lemma.

Lemma 2a.2. *For every path α and all points $a, b, c \in I$ we have*

$$\alpha_{a:b} \star \alpha_{b:c} \simeq_P \alpha_{a:c}.$$

Proof. A path homotopy between $\alpha_{a:b} \star \alpha_{b:c}$ and $\alpha_{a:c}$ is the map H defined by $H(\cdot,t) = \alpha_{a:h(t)} \star \alpha_{h(t):c}$ where $h(t) = (1-t)b + t(a+c)/2$. □

The relation of path homotopy is an equivalence relation, as one can check directly. To prove that $\alpha \simeq_P \beta$ implies $\beta \simeq_P \alpha$, for example, it suffices to note that if F is a path homotopy between α and β, then the map $(s,t) \mapsto F(s, 1-t)$ is a path homotopy between β and α. The equivalence class of a path α under path homotopy is denoted $[\alpha]_P$, and is called the **path class** of α.

The fundamental group

We now define a concatenation operation for path classes. Path concatenation respects path homotopy, in the sense that if $\alpha \simeq_P \gamma$ and $\beta \simeq_P \delta$, then $\alpha\star\beta \simeq_P \gamma\star\delta$. Thus the concatenation $[\alpha]_P \star [\beta]_P$ of two path classes is well defined by setting $[\alpha]_P \star [\beta]_P = [\alpha \star \beta]_P$. The important properties of this operation are listed below; they can be derived from Lemma 2a.2.

(1) Associativity: $([\alpha]_P \star [\beta]_P) \star [\gamma]_P = [\alpha]_P \star ([\beta]_P \star [\gamma]_P)$ whenever these expressions are defined.

(2) Existence of identities: $[\alpha]_P \star [\alpha_{1:1}]_P = [\alpha]_P = [\alpha_{0:0}]_P \star [\alpha]_P$ for any path class $[\alpha]_P$. Thus $[\alpha_{1:1}]_P$ and $[\alpha_{0:0}]_P$ are right and left identities, respectively, for $[\alpha]_P$.

(3) Existence of inverses: $[\alpha]_P \star [\widehat{\alpha}]_P = [\alpha_{0:0}]_P$ and $[\widehat{\alpha}]_P \star [\alpha]_P = [\alpha_{1:1}]_P$ for any path class $[\alpha]_P$. Thus $[\widehat{\alpha}]_P$ is both a left and right inverse for $[\alpha]_P$; its own inverse is $[\alpha]_P$, since the reverse of $\widehat{\alpha}$ is α.

Equations (1) through (3) are called the **groupoid properties** of concatenation. They would make concatenation a group operation, except that the concatenation of two paths is not always defined. To obtain a group we need only restrict ourselves to paths that begin and end at a specific point.

Definition 2a.3. Let x_0 be a point of the space X. A path in X whose endpoints coincide at x_0 is called a **loop** at x_0. The **fundamental group** of X at x_0, denoted $\pi_1(X, x_0)$, is the set of path classes of loops at x_0, under the operation of concatenation.

The identity element of $\pi_1(X, x_0)$ is the class $[t \mapsto x_0]_P$ of the constant loop at x_0. A loop at x_0 is called **inessential** if it falls in this class, and **essential** otherwise.

A natural question to ask is how the fundamental group $\pi_1(X, x_0)$ depends on the choice of base point x_0. For a path-connected sךace, the answer is that the fundamental groups at different base points are isomorphic. To see why, let α be a path in X from x to y, and consider the map $h_\alpha\colon \pi_1(X,x) \to \pi_1(X,y)$ defined by

$$h_\alpha([\gamma]_P) = [\widehat{\alpha} \star \gamma \star \alpha]_P.$$

This map is a group homomorphism; one simply computes, using the groupoid properties of concatenation, that

$$\begin{aligned} h_\alpha([\gamma]_P \star [\delta]_P) &= [\widehat{\alpha} \star (\gamma \star \delta) \star \alpha]_P \\ &= [(\widehat{\alpha} \star \gamma \star \alpha) \star (\widehat{\alpha} \star \delta \star \alpha)]_P \\ &= h_\alpha([\gamma]_P) \star h_\alpha([\delta]_P). \end{aligned}$$

Furthermore, if $\beta = \widehat{\alpha}$, then h_β and h_α are inverses, so h_α is actually an isomorphism.

Definition 2a.4. A space X is **simply connected** if X is path-connected and $\pi_1(X, x_0)$ equals 0, the trivial group, for some point $x_0 \in X$.

Because all the fundamental groups of a path-connected space are isomorphic, a simply connected space has trivial fundamental group at every point. In other words, every loop in a simply connected space is inessential. As a consequence we deduce a very useful property of simply connected spaces.

Lemma 2a.5. *In a simply connected space, any two paths having the same initial and final points are path-homotopic.*

Proof. Let X be a simply connected space, and let α and β be two paths in X from x to y. Then $\beta \star \widehat{\beta}$ is a loop at x, and because $\pi_1(X, x)$ is trivial, we have $\beta \star \widehat{\alpha} \simeq_P \beta_{0:0}$. Hence by the groupoid properties of concatenation,

$$[\beta]_P = [\beta \star \beta_{1:1}]_P = [\beta \star (\widehat{\alpha} \star \alpha)]_P = [(\beta \star \widehat{\alpha}) \star \alpha]_P = [\beta_{0:0} \star \alpha]_P = [\alpha]_P.$$

Therefore β is path-homotopic to α. □

Induced homomorphisms

Not only can we associate with each space a fundamental group, but to each map between spaces we can associate a homomorphism between the corresponding fundamental groups. Suppose $f\colon X \to Y$ is a map of topological spaces, and suppose $f(x_0) = y_0$. We usually express this fact by writing $f\colon (X, x_0) \to (Y, y_0)$. Then f **induces** a homomorphism

$$f_\star\colon \pi_1(X, x_0) \to \pi_1(Y, y_0),$$

defined by $f_\star([\alpha]_P) = [f \circ \alpha]_P$. This map is well defined, because if α is a loop at x_0, then $f \circ \alpha$ is a loop at y_0; if β is another path in X, and H is a path homotopy between α and β, then $f \circ H$ is a path homotopy between the paths $f \circ \alpha$ and $f \circ \beta$. The map $f_\star$ is a group homomorphism because $f \circ (\alpha \star \beta) = (f \circ \alpha) \star (f \circ \beta)$, which implies

$$f_\star([\alpha \star \beta]_P) = f_\star([\alpha]_P) \star f_\star([\beta]_P).$$

The correspondence between maps of spaces and homomorphisms of fundamental groups is actually a "functor", which means that it has the following **functorial properties**:

(1) It commutes with composition, that is, $(g \circ f)_* = g_* \circ f_*$.

(2) It takes identity maps to identity maps, so $id_* = id$.

One important consequence of these properties is that the fundamental group of a space is a **topological invariant**, meaning that homeomorphisms preserve it. For suppose that $f: (X, x_0) \to (Y, y_0)$ is a homeomorphism with inverse $g: (Y, y_0) \to (X, x_0)$. Then the maps g_* and f_* are inverses: by property (1), we have $f_* \circ g_* = (f \circ g)_* = id_*$, which by property (2) is the identity homomorphism on $\pi_1(Y, y_0)$; similarly $g_* \circ f_* = (g \circ f)_*$ is the identity on $\pi_1(X, x_0)$. Therefore f_* gives an isomorphism between the fundamental groups of X (at x_0) and Y (at y_0).

Similar reasoning shows that if A is a retract of X, and $x_0 \in A$, then the map $i_*: \pi_1(A, x_0) \to \pi_1(X, x_0)$ induced by the inclusion $i: (A, x_0) \to (X, x_0)$ is a monomorphism (one-to-one). For if $r: (X, x_0) \to (A, x_0)$ is the retraction, then $r \circ i = id_A$, whence $r_* \circ i_*$ is the identity on $\pi_1(A, x_0)$. Since $Ker\, i_* \subseteq Ker(r_* \circ i_*) = 0$, the kernel of i_* is trivial. As a corollary, every retract of a simply connected space is simply connected.

Homotopy of general maps

There are many types of homotopy relations, path homotopy being only one of them. As we are concerned primarily with homotopy among paths and loops, the following results will be used mainly for proving spaces to be simply connected.

Definition 2a.6. Let X and Y be topological spaces, and let $A \subseteq X$. Two maps $f, g: X \to Y$ are **homotopic relative to A**, written $f \simeq g \operatorname{rel} A$, if there is a map $F: X \times I \to Y$ such that $F(\cdot, 0) = f$, $F(\cdot, 1) = g$, and $F|_{A \times I} = id_{A \times I}$. If $A = \emptyset$, we simply write $f \simeq g$. The map F is a **homotopy** between f and g.

Though the concept of homotopy seems to apply only to maps, it can tell us something about a space when applied to the identity map on the space. A subspace A of a space X is a **deformation retract** if there is a retraction $r: X \to A$ such that $id_X \simeq i \circ r \operatorname{rel} A$, where $i: A \to X$ is the inclusion map. The homotopy between r and id_X is called a **deformation retraction**. The fundamental group of a deformation retract satisfies an even stronger property than that of a retract.

Lemma 2a.7. *If A is a deformation retract of X, then the inclusion $i: (A, x_0) \to (X, x_0)$ induces an isomorphism of fundamental groups.*

Proof. Because A is a retract of X, the map $i_*: \pi_1(A, x_0) \to \pi_1(X, x_0)$ is a monomorphism. It remains to show that i_* is an epimorphism (onto). Let β be a loop at

x_0; we prove that $[\beta]_P$ is in the image of i_* by applying the deformation retraction to β. Let $F\colon X \times I \to X$ be a deformation retraction of X to A, and define a map $G\colon I \times I \to X$ by $G(s,t) = F(\beta(s),t)$. Then G is a path homotopy, since for $e \in \{0,1\}$, the point $G(e,t)$ is $F(\beta(0),t) = F(x_0,t) = x_0$ (because F is the identity on $A \times I$). Moreover, F is a homotopy between β and a loop $\alpha\colon I \to X$ whose image lies in A: we have $G(\cdot\,,0) = F(\alpha(\cdot),0) = \beta$, and $G(\cdot\,,1) \subseteq F(X,1) \subseteq A$. Therefore $\beta \simeq_P \alpha$. Let $\alpha'\colon I \to A$ be the path $t \mapsto \alpha(t)$ in A. Then $[\alpha']_P \in \pi_1(A,x_0)$, and $i_*([\alpha']_P) = [i \circ \alpha']_P = [\alpha]_P$. Since $[\alpha]_P = [\beta]_P$, this means $[\beta]_P \in Im\, i_*$. □

Lemma 2a.7 gives us one way to show that a space is simply connected. Say that a space X is **contractible** if some point of X is a deformation retract of X. As an example, any starlike or convex subset of R^n is contractible. For if $X \subseteq R^n$ contains a point z such that the line segment $\overline{xz}$ lies in X whenever x does, then the map $F\colon X \times I \to X$ defined by $F(x,\cdot) = x \triangleright z$ is a deformation retraction of X to z. We call it a **contraction** of X to z.

Lemma 2a.8. *Every contractible space is simply connected.*

Proof. Let $F\colon X \times I \to X$ be a contraction of X to the point $z \in X$. Then X is path-connected because any point $x \in X$ can be joined to z by the path $\rho_x = F(x,\cdot)$; for any two points $x, y \in X$, the concatenation $\rho_x \star \widehat{\rho_y}$ is a path between x and y. Because z is a deformation retract of x, the previous lemma shows that X and z have isomorphic fundamental groups. There is only one path in z, so $\pi_1(z,z)$ is trivial. Hence $\pi_1(X,z) = 0$ also. □

Extension lemma

We conclude the section with a criterion for a loop to be inessential.

Lemma 2a.9. *Let $f\colon Fr(I \times I) \to X$ be any map, and let δ be the loop*

$$\delta = (\cdot\,,0) \star (1,\cdot\,) \star \widehat{(\cdot\,,1)} \star \widehat{(0,\cdot\,)}\colon I \to I \times I.$$

The loop $f \circ \delta$ is inessential if and only if f has an extension $F\colon I \times I \to X$. □

2B. Covering Spaces

In order to compute the fundamental groups of spaces that are not simply connected, one usually introduces the notion of a covering space. As we shall see, the fundamental group of the circle S^1 is easily determined using this device. But I introduce covering spaces for a different reason. In essence, a simply connected covering space provides a spatial representation of path homotopy classes. It thereby converts problems involving homotopy constraints, such as my single-layer wire routing problems, into problems without homotopy constraints.

This section provides a very brief introduction to the theory of covering spaces. It defines covering spaces and the notion of *lifting* to a covering space, and proves the important theorem that the lifting of a map is unique if the lifting is determined at a single point. It then gives conditions for a map to be liftable, and notes that paths and homotopies of paths can always be lifted. Some applications of lifting are also presented. Finally, it states some fairly mild conditions under which a space has a simply connected covering space, and shows that in the presence of those conditions, that covering space is essentially unique.

Definition of covering space

Definition 2b.1. Let $p: M \to X$ be a surjective map. An open set U in X is **evenly covered** by p if $p^{-1}(U)$ can be partitioned into disjoint open sets, each of which is mapped homeomorphically onto U by p. If every point of X has a neighborhood that is evenly covered by p, then p is called a **covering map**, and M a **covering space** of X.

A covering space is often called simply a **cover**; the space it covers is called the **base** space. Locally, a covering space looks like a union of disjoint copies of the base. It follows immediately that a covering map is a local homeomorphism. For suppose that $p: M \to X$ is a covering map, and let v be a point of M. Take U to be a neighborhood of $p(v)$ that is evenly covered by p, and partition $p^{-1}(U)$ into disjoint open sets that are mapped homeomorphically onto U by p. One of these open sets, call it V, contains v. Then V is a neighborhood of v, and $p|_V$ is a homeomorphism onto its image, which is open. This makes p a local homeomorphism. As a consequence, M has all the local properties that X has.

Perhaps the simplest interesting covering map is $\theta: R^1 \to S^1$ given by

$$\theta(t) = (\cos 2\pi t, \sin 2\pi t),$$

which maps the real line onto the circle. We show that every point of S^1 is evenly covered by θ. Let s_0 represent the point $(1,0)$ of S^1. Then $S^1 - s_0$ is a neighborhood of every point of S^1 but s_0, and is evenly covered by θ. For $\theta^{-1}(S^1 - s_0)$ is $R^1 - Z$, which is the disjoint union of the open intervals $\{(n, n+1) : n \in Z\}$, and each of these intervals is mapped homeomorphically onto $S^1 - s_0$ by θ. Similarly, the neighborhood $S^1 - (-1,0)$ of s_0 is evenly covered by θ. Therefore θ is a covering map. A related covering map, pictured in Figure 2b-1, is $\theta \times id_I: R^1 \times I \to S^1 \times I$. This map can be thought of as compressing an infinite helical surface in R^3 onto an annular region in R^2.

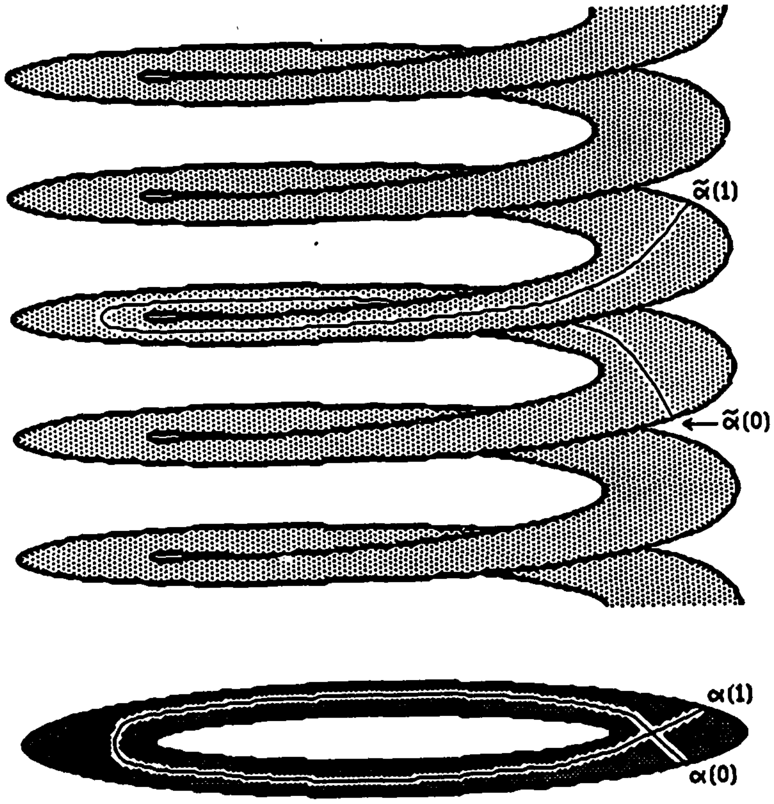


Figure 2b-1. *Lifting to a covering space.* The helical surface, which extends infinitely in both directions, is a simply connected covering space for the annular region below. The covering map is downward projection. Also shown is a path α in the annulus and one of its liftings $\widetilde{\alpha}$ to the covering space. There is one such lifting for each point in the inverse image of $\alpha(0)$.

Lifting

One can study objects in a base space by transporting those objects to some covering space. If $p: M \rightarrow X$ is a covering map, and $g: Y \rightarrow X$ is a map into X, a **lifting**, or **lift**, of g is a map $\widetilde{g}: Y \rightarrow M$ satisfying $p \circ \widetilde{g} = g$. For example, if g is a path in X, then $\widetilde{g}$ is a path in M that "sits over" g. (See Figure 2b-1.)

Theorem 2b.2. (Uniqueness of Liftings) *Two liftings of a map from a connected space are equal if they agree at one point.*

Proof. Let $p: M \rightarrow X$ be a covering map, and let $g, g': Y \rightarrow M$ be two liftings of a map $f: Y \rightarrow X$. Let $Y_=$ be the set of points in Y at which g and g' agree, and let $Y_{\neq}$ be its complement. To prove the theorem, it suffices to show that $Y_=$ and $Y_{\neq}$ are both open in Y. For if Y is connected, it follows that either $Y_=$ or $Y_{\neq}$ is empty.

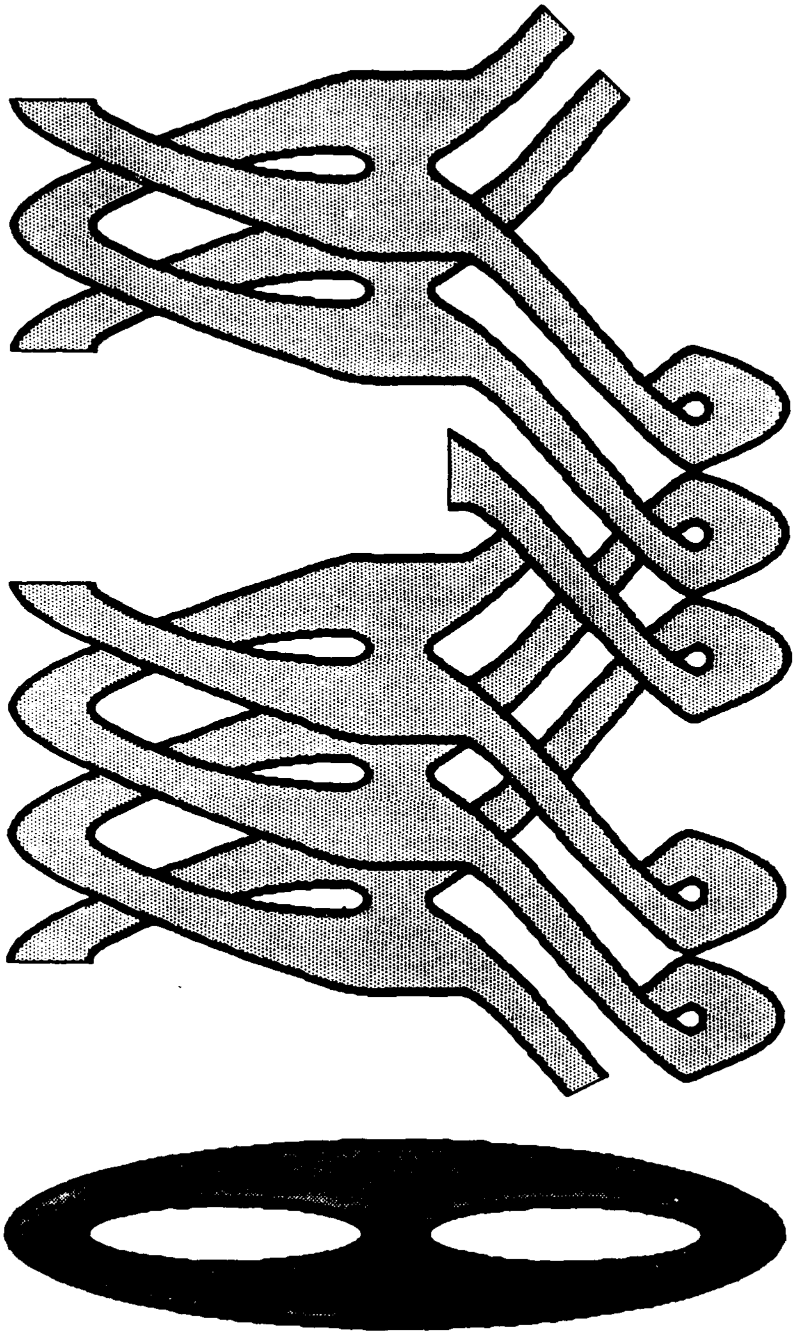


Figure 2b-2. *A complicated covering space.* The surface pictured above is part of the simply connected covering space for a disk with two circular holes removed. The "layers" of this covering space are indexed by the free group on two generators (which is also the fundamental group of the base space). Only a few layers are shown.

Let y be a point of Y, and choose a neighborhood $U \subseteq X$ of $f(y)$ that is evenly covered by p. Because f is continuous, there is a neighborhood V of y such that $f(V) \subseteq U$. We may assume that V is connected. Now let W be the component of $p^{-1}(U)$ that contains $g(y)$. Because $g(V)$ is connected, and $g(V) \subseteq p^{-1} \circ f(V) \subseteq p^{-1}(U)$, the set $g(V)$ must lie entirely within W. Similarly, if W' is the component of $p^{-1}(U)$ that contains $g'(y)$, then $g'(V) \subseteq W'$. If $y \in Y_{\neq}$, then $W \cap W' = \emptyset$, so $g(V) \cap g'(V) = \emptyset$ and therefore V is a neighborhood of y in $Y_{\neq}$. If instead $y \in Y_{=}$, then $W = W'$, whence $g(v) = g'(v)$ for all $v \in V$ because $p \circ g = p \circ g'$ and $p|_W$ is injective. In this case, V is a neighborhood of y in $Y_{=}$. Thus $Y_{=}$ and $Y_{\neq}$ are both open in Y. □

A natural question is: When can a map be lifted to a covering space? The following theorem gives a complete answer to this problem for a large class of spaces. Recall that a space is **locally path-connected** if it has a basis of path-connected sets. For example, any convex subspace $X \subseteq R^n$ is locally path-connected, because every open ball in X is convex and hence path-connected.

Theorem 2b.3. (Lifting Theorem) *Let $p: (M, m_0) \to (X, x_0)$ be a covering map, and let Y be connected and locally path-connected. The map $g: (Y, y_0) \to (X, x_0)$ has a lifting $\widetilde{g}: (Y, y_0) \to (M, m_0)$ if and only if*

$$g_*(\pi_1(Y, y_0)) \subseteq p_*(\pi_1(M, m_0)). \quad \square$$

In particular, the conclusion always holds if Y is simply connected and locally path-connected, for then Y is connected and $Im\, g_*$ is trivial. For each point $m_0 \in p^{-1}(x_0)$, the map $g: (Y, y_0) \to (X, x_0)$ can then be lifted in such a way that $\widetilde{g}(y_0) = m_0$. In particular, every path $\alpha: I \to X$ and every path homotopy $F: I \times I \to X$ can be so lifted: the spaces I and $I \times I$ are convex, and hence locally path-connected, contractible, and simply connected (Lemma 2a.8). Actually, the proof of the lifting theorem requires these facts, and the following proposition as well. The proof of the proposition, although it relies on the lifting of path homotopies, is nevertheless instructive.

Proposition 2b.4. *Liftings of homotopic paths are path-homotopic if they agree at one endpoint.*

Proof. Let $p: M \to X$ be a covering map, and let $\widetilde{\alpha}$ and $\widetilde{\gamma}$ be paths in M whose projections $\alpha = p \circ \widetilde{\alpha}$ and $\gamma = p \circ \widetilde{\gamma}$ are path homotopic. Let $F: I \times I \to X$ be a path homotopy between α and γ, and suppose that $\widetilde{\alpha}(e) = \widetilde{\gamma}(e)$ where e is either 0 or 1. Choose the base points $(e, 0)$, $\alpha(e)$, and $\widetilde{\alpha}(e)$ for $I \times I$, X, and M respectively. Then F lifts to a map $\widetilde{F}: I \times I \to M$ satisfying $\widetilde{F}(e, 0) = \widetilde{\alpha}(e)$. I claim that $\widetilde{F}$ is a path homotopy between $\widetilde{\alpha}$ and $\widetilde{\gamma}$.

- $\widetilde{F}(e,\cdot)$ *is constant.* Both $t \mapsto \widetilde{F}(e,t)$ and $t \mapsto \widetilde{\alpha}(e)$ are liftings of the constant path $t \mapsto F(e,t)$, and they agree at $t = 0$. The interval I is connected, so by uniqueness of liftings (Theorem 2b.2), the liftings are identical. In particular, $\widetilde{F}(e,1) = \widetilde{\alpha}(e) = \widetilde{\gamma}(e)$.
- $\widetilde{F}(\cdot,0) = \widetilde{\alpha}$. Both $\widetilde{\alpha}$ and $\widetilde{F}(\cdot,0)$ lift α, because $p \circ \widetilde{F}(\cdot,0) = F(\cdot,0) = \alpha$, and the two liftings agree at e. By (2b.2) again, they must be the same map.
- $\widetilde{F}(\cdot,1) = \widetilde{\gamma}$. Both $\widetilde{\gamma}$ and $\widetilde{F}(\cdot,1)$ lift γ, and they agree at e.
- $\widetilde{F}(1-e,\cdot)$ *is constant.* Both $t \mapsto \widetilde{F}(1-e,t)$ and $t \mapsto \widetilde{F}(1-e,0)$ lift the constant path $t \mapsto F(1-e,\cdot)$. Because the liftings agree when $t = 0$, they must be equal. □

Armed with this lemma and the covering map $\theta\colon (R^1,0) \to (S^1,s_0)$, the reader should be able to prove the following result. (Hint: lift each loop α at s_0 to a path $\widetilde{\alpha}$ beginning at 0, and consider $\widetilde{\alpha}(1)$.)

Proposition 2b.5. *The fundamental group of the circle is isomorphic to the integers under addition.*

Existence and uniqueness of covering spaces

What makes the proof of Proposition 2b.5 work is that the circle has a simply connected covering space (the real line) with a natural group operation (addition). Most spaces do not come with as nice a covering space as R^1. Nevertheless, simply connected covering spaces can often be constructed out of the space of paths in the base space. The following theorem gives sufficient conditions for this construction to work. The conditions may look scary, but in fact they are satisfied by almost every decent space. We say that a space X is **semilocally simply connected** if every point $x \in X$ has a neighborhood U such that the map $i_*\colon \pi_1(U,x) \to \pi_1(X,x)$ induced by the inclusion $i\colon U \to X$ is trivial. Of course, this condition holds if U is simply connected.

Theorem 2b.6. *Every connected, locally path-connected, semilocally simply connected space has a simply connected covering space.* □

There is a notion of equivalence among covering spaces of a fixed base space. This notion is stronger than that of homeomorphism, because it also requires that the correspondence between the covering spaces respect the covering maps. More specifically, if $p\colon M \to X$ and $q\colon N \to X$ are covering maps, then M and N are **equivalent** if there are inverse maps $f\colon M \to N$ and $g\colon N \to M$ such that $q \circ f = p$ and $p \circ g = q$. Equivalent covering spaces are topologically indistinguishable. The following proposition shows that the simply connected cover of a decent space is essentially unique.

Proposition 2b.7. *All simply connected covering spaces of a connected, locally path-connected space are equivalent.*

Proof. (Lift the covering maps.) Let X be connected and locally path-connected, let $p: (M, m_0) \rightarrow (X, x_0)$ and $q: (N, n_0) \rightarrow (X, x_0)$ be covering maps, and suppose that M and N are simply connected. The maps p and q are local homeomorphisms, so M and N have all the local properties that X has. In particular, M and N are locally path-connected. We now apply the Lifting Theorem (2b.3), lifting p to a map $\widetilde{p}: (M, m_0) \rightarrow (N, n_0)$, and also lifting q to a map $\widetilde{q}: (N, n_0) \rightarrow (M, m_0)$. By the definition of lifting, $p \circ \widetilde{q} = q$ and $q \circ \widetilde{p} = p$. I claim that $\widetilde{p}$ and $\widetilde{q}$ are inverses, making M and N homeomorphic. Because $p \circ \widetilde{q} \circ \widetilde{p} = q \circ \widetilde{p} = p$, we find that $\widetilde{q} \circ \widetilde{p}$ is a lift of the map p. But id_M also lifts p, and because $\widetilde{q} \circ \widetilde{p}(x_0) = \widetilde{q}(y_0) = x_0$, the two maps agree at the point x_0. Since M is connected, they must be identical, by Theorem 2b.2. Entirely symmetrical reasoning shows that $id_Y = \widetilde{p} \circ \widetilde{q}$. □

Covering transformations

Proposition 2b.7 not only shows that the simply connected cover of a decent space is unique, but also implies that this cover must be highly symmetrical. If $p: M \rightarrow X$ is a covering map, a homeomorphism $h: M \rightarrow M$ that lifts p is called a **covering transformation** of M. Suppose that M is simply connected and X is locally path-connected. For any two points $m_0, m_1 \in M$ that have the same image x_0 under p, Proposition 2b.7 gives us a covering transformation $h: M \rightarrow M$ such that $h(m_0) = m_1$. (Consider the covering maps $p: (M, m_0) \rightarrow (X, x_0)$ and $p: (M, m_1) \rightarrow (X, x_0)$.) Hence different lifts of the same path or homotopy are related by a covering transformation. This fact allows us to ignore the base point of the covering space; all base points are equivalent.

We conclude this section with another simple application of the Lifting Theorem. It shows how one can lift subspaces as well as maps.

Lemma 2b.8. *Let $p: M \rightarrow X$ be a covering map, and let C be a simply connected, locally path-connected subspace of X. For every path component A of $p^{-1}(C)$, the map $p: A \rightarrow C$ is a homeomorphism.*

Proof. Let c be a point of C, and pick $a \in A \subseteq p^{-1}(C)$. Lift the identity map on C to a map $i: (C, c) \rightarrow (p^{-1}(C), a)$. Because C is path-connected, so is $i(C)$, and hence $i(C) \subseteq A$. We have $p \circ i = id_C$, and it remains to show $i \circ p = id_A$. The map $i \circ p: (A, a) \rightarrow (A, a)$ lifts $p: (A, a) \rightarrow (C, c)$, because $p \circ (i \circ p) = (p \circ i) \circ p = id_C \circ p$. Hence id_A and iop are two liftings of $p: A \rightarrow C$, and they satisfy $id_A(a) = a = iop(a)$. Therefore $id_A = i \circ p$ by uniqueness of liftings. □

2C. Paths and Loops in the Plane

This section collects miscellaneous results concerning the topology of subsets of the plane. Most are of the "obvious but nontrivial" variety, like the fact that a simple loop is inessential in a subspace of the plane if and only if that subspace includes the inside of the loop. These results can be justified using standard topological methods. One result, however, we derive from a theorem of real analysis. Our result says: Within any nonempty family of canonical paths in a bounded subspace of tha plane, there exists a sequence of paths converging to a path whose euclidean arc length is no greater than that of any path in the family. We use this result in proving the existence of things like rubber bands and ideal routes, which are defined as the minimum-length paths in certain families.

Facts about polygons

Geometric topology, the study of topology within euclidean spaces, is another source of mathematical insight into single-layer wire routing problems. In these problems the routing region—a subspace of the plane—has important topological properties that one often takes for granted. Perhaps the most famous of these is the Jordan Curve Theorem, stated below for the case of piecewise linear loops.

Theorem 2c.1. *(Jordan Curve Theorem) If λ is a simple loop in R^2, then $R^2 - Im\,\lambda$ has two components, one bounded and one unbounded, whose common frontier is $Im\,\lambda$.* □

The bounded component is called the **inside** of the loop, and the other component is called the **outside**. A kind of converse to Theorem 2c.1 is the following.

Lemma 2c.2. *Let $X \subset R^2$ be a finite union of polygonal regions. If A is a bounded component of $R^2 - X$, then there is a simple loop in X whose inside contains A.* □

We shall also need the following results.

Proposition 2c.3. *If λ is a simple locp in R^2, then $Im\,\lambda$ is a retract of $R^2 - inside(\lambda)$.* □

Proposition 2c.4. *Let λ and μ be simple loops in R^2. If $Im\,\mu \subset inside(\lambda)$, then $Im\,\mu$ is a deformation retract of $R^2 - inside(\mu) - outside(\lambda)$.* □

Theorems like these belong to geometric topology, and are somewhat messy to prove rigorously, even when stated (as here) for piecewise linear objects only. Unfortunately, I have no reference for these particular results, though they follow from well-known properties of polygons. One reference for geometric topology is [35].

Enclosure

Another intuitive property of planar loops is this: A simple loop λ whose inside contains a point p cannot be deformed so that p ends up outside, except by crossing over p. More formally, if p lies inside λ, and if the simple loop μ is loop-homotopic to λ in $R^2 - p$, then p lies inside μ.

We can obtain a more general result by extending the notion of "inside" to loops that are not simple. Say a loop λ in R^2 **encloses** a connected subset F of $R^2 - Im\,\lambda$ if λ is essential in $R^2 - F$. Now suppose $\lambda \simeq_P \mu$ as paths in $R^2 - F$. Then either both loops are essential or both are inessential in $R^2 - F$, so λ encloses F if and only if μ does. The next proposition shows that the definition of enclosure agrees with the definition of inside for simple loops.

Proposition 2c.5. *Let λ be a loop in R^2, and let S denote the space $R^2 - Im\,\lambda$. If λ encloses a connected subset F of S, then F lies in a bounded component of S. The converse holds if λ is simple.* □

As a consequence we obtain a very intuitive result concerning simple loops.

Corollary 2c.6. *A simple loop λ is inessential in a subspace S of R^2 if and only if S includes the inside of λ.*

Proof. Suppose that S includes *inside*(λ), and let F denote the outside of λ. By Proposition 2c.5, λ does not enclose F, which means λ is inessential in $R^2 - F$. Hence λ is inessential in the larger space S. Now suppose that S does not include *inside*(λ), and let F be a component of $R^2 - S$ lying inside λ. By Proposition 2c.5 again, λ encloses F, which means that λ is essential in $R^2 - F$. Hence λ is essential in the smaller space S. □

An elementary property of enclosure is the following: If λ and μ are loops based at the same point, and λ does not enclose F, then $\lambda \star \mu$ encloses F if and only if μ does. More generally, suppose $\lambda_1, \ldots, \lambda_n$ are loops based at the same point, and let F be a connected set that intersects none of them. The concatenated loop $\lambda_1 \star \cdots \star \lambda_n$ can enclose F only if some λ_i does, and it *does* enclose F if exactly one λ_i does.

Piecewise linearity

When dealing with piecewise linear paths, we shall often want our homotopies to be piecewise linear also. A map F from $I \times I$ into a linear space is **piecewise linear** if $I \times I$ can be divided into triangles so that F is linear on each triangle. If F is piecewise linear and α is any linear path in $I \times I$, then $F \circ \alpha$ is a piecewise linear path. The following lemma allows us to assume, in many cases, that our homotopies are piecewise linear.

Lemma 2c.7. *Let $S \subset R^2$ be the union of finitely many triangles, and let $F: I \times I \to S$ be a homotopy. If the paths $F(0,\cdot)$, $F(1,\cdot)$, $F(\cdot,0)$, and $F(\cdot,1)$ are piecewise linear, then there is a piecewise linear map $G: I \times I \to S$ that agrees with F on $Fr\, I \times I$.*

Outline of proof. (For those familiar with simplicial complexes.) One first constructs a triangulation of S and a triangulation of $I \times I$ such that the map $F|_{Fr\, I \times I}$ is simplicial. Then one applies the Simplicial Approximation Theorem to F. The result is a piecewise linear map G that agrees with F on $Fr\, I \times I$. $\square$

Minimization of arc length

One idea that was already put to use in Chapter 1 is the construction of the minimum-length path satisfying some condition. The following proposition gives us a tool for constructing a minimum-length path as a limit of other paths. It relies on a classical theorem from topology and real analysis called Ascoli's Theorem.

Proposition 2c.8. *Let Λ be a nonempty family of canonical paths in a bounded subspace S of R^2, and put $l = \inf\{\,|\lambda| : \lambda \in \Lambda\,\}$. Then Λ includes a uniformly convergent sequence of paths whose limit has euclidean arc length at most l.*

Proof. Let $\Delta = \langle \delta_n \rangle_{n=1}^{\infty}$ be a sequence of paths in Λ whose euclidean arc lengths converge to l. We use Ascoli's theorem, taken from [46], to show that the sequence $\langle \delta_n \rangle$ has a convergent subsequence.

Definition: Let Φ be a family of functions from a space X to a metric space Y with metric σ. The family Φ is **equicontinuous** if for every point $x \in X$ and every $\epsilon > 0$, there is a neighborhood N of x such that $\sigma[f(x), f(y)] < \epsilon$ for all $y \in N$ and all $f \in \Phi$.

Theorem: (Ascoli's Theorem) *Let Φ be an equicontinuous family of functions from a separable space X to a metric space Y. Let $\langle f_n \rangle$ be a sequence in Φ such that for each $x \in X$ the closure of the set $\{\, f_n(x) : n > 0\,\}$ is compact. Then there is a subsequence $\langle f_{n_k} \rangle$ that converges pointwise to a continuous function f, and the convergence is uniform on each compact set of X.*

In our case, the family of functions is Δ, the space X is the unit interval I, and the space Y is S with the euclidean metric. We check the conditions of Ascoli's Theorem in order. Let u be a bound on the arc lengths of the paths δ_i. The family Δ is equicontinuous, because if $x \in I$ and $\epsilon > 0$ and $\delta \in \Delta$, every point y in the open set $I \cap (x - \epsilon/u, x + \epsilon/u)$ satisfies

$$|\delta(x) - \delta(y)| \le |\delta_{x:y}| = |y - x| \cdot |\delta| < (\epsilon/u)\ |\delta| \le \epsilon.$$

The space I is separable because the set of rationals in I is countable and dense in I. Finally, the set $\{\, \delta_n(x) : n > 0\,\}$ lies in the bounded set S, which implies that its closure is compact.

We conclude that Ascoli's Theorem is applicable to the sequence $\langle \delta_n \rangle$. It yields a subsequence $\langle \alpha_k \rangle$ of $\langle \delta_n \rangle$ that converges to a path α. Because I is compact, the convergence is uniform.

It remains to show that $|\alpha| \leq l$. Let γ be any piecewise linear approximation to α, and let ϵ be any positive real number. The lengths of the paths α_k converge to l, so there is number K such that for all $k > K$, we have $|\alpha_k| < l + \epsilon$. Suppose γ has m segments. Because the functions α_k converge uniformly to α, for all sufficiently large k we have $|\alpha_k(t) - \alpha(t)| < \epsilon/m$ for all $t \in I$. In particular, if the ith vertex of γ is $\gamma(t_i) = \alpha(t_i)$, we may choose k so that $|\alpha_k(t_i) - \alpha(t_i)| < \epsilon/m$ for all i. The points $\alpha_k(t_i)$ divide α_k into small pieces that correspond to the segments of γ. The length of the piece from $\alpha_k(t_i)$ to $\alpha_k(t_{i+1})$ is at least the length of the corresponding segment of γ, less $2\epsilon/m$. Summing these inequalities, we find that

$$|\gamma| \leq |\alpha_k| + 2\epsilon < l + 3\epsilon.$$

Because ϵ was arbitrary, it follows that every piecewise linear approximation to α has length l or less. Therefore the arc length of α is at most l. □

2D. Topological Manifolds

Almost all the spaces dealt with in this paper are manifolds, with or without boundary. A manifold is a very nice kind of topological space; it looks locally like R^n or H^n. This section establishes the properties of manifolds that will be needed later on.

Definition 2d.1. If x is a point in a space X, a **patch** about x is a homeomorphism of a neighborhood of x with an open set of H^n. An ***m*-manifold with boundary** is a nonempty Hausdorff space in which every point has a patch. The **boundary** of an m-manifold with boundary is the set of points x having a patch h such that $h(x) \in R^{m-1} \subset H^m$. Such a patch is called a **boundary patch**.

I will always use the term ***m*-manifold** to mean m-manifold with boundary. The boundary of an m-manifold M, which is an $(m-1)$-manifold, is denoted $Bd\,M$. A classical theorem [39, p. 207] shows that if $x \in M$ has a boundary patch, then every patch about x is a boundary patch.

Theorem 2d.2. (Invariance of Domain) *Let $U \subseteq R^n$ be open, and let $f: U \to R^n$ be continuous and injective. Then $f(U)$ is open in R^n and f is an embedding.* □

To see how this theorem applies to manifolds, let $h: U \to V$ and $h': U' \to V'$ be two patches about the same point x in an m-manifold. If h is not a boundary patch, then there is a neighborhood W of $h(x)$ in V that is open in R^m. The map $h' \circ h^{-1}|_W$

that sends W into H^m is continuous and injective, and hence by Theorem 2d.2, its image is open in R^m. But this image contains $h'(x)$, and therefore $h'(x)$ does not lie in R^{m-1}. Thus h' is not a boundary patch.

One can infer that $M - Bd\,M$ is open in M, for any manifold M. For if $x \in M - Bd\,M$, there is a patch $h: U \to V$ about x whose image does not intersect R^{m-1}. This homeomorphism h is a nonboundary patch for every point in U, and hence no point of U lies in $Bd\,M$. But U is a neighborhood of x. Therefore $M - Bd\,M$ is open. Note that H^m itself is an m-manifold, and $Bd\,H^m = R^{m-1}$.

Facts about manifolds

Our first lemma concerns covering spaces of manifolds. If $f: M \to X$ is a map from a manifold M, then $Bd\,f$ denotes the restriction of f to $Bd\,M$.

Lemma 2d.3. *If $p: M \to X$ is a covering map and X is an m-manifold, then M is an m-manifold and $Bd\,p: Bd\,M \to Bd\,X$ is also a covering map.*

Proof. Consider any point v of M. Because p is a local homeomorphism, v has a neighborhood V that is mapped homeomorphically by p onto a neighborhood U of $p(v)$ Choose a patch $h: U' \to h(U')$ about $p(v)$. Then p maps the neighborhood $V \cap p^{-1}(U')$ of v homeomorphically onto $U \cap U'$, which itself is homeomorphic under h to the open set $h(U \cap U')$ of H^m. Hence $h \circ p$, when restricted to $V \cap p^{-1}(U)$, is a patch about v. Therefore M is an m-manifold. Furthermore, if $p(v) \in Bd\,X$, then $h \circ p$ maps v to a point of $Bd\,H^m$, so v lies in $Bd\,M$; conversely, if $p(v) \notin Bd\,X$, then $h \circ p(v) \notin Bd\,H^m$, so $v \notin Bd\,M$. Therefore p maps $Bd\,M$ onto $Bd\,X$.

It remains to show that the surjection $Bd\,p: Bd\,M \to Bd\,X$ is a covering map. Let x be a point of $Bd\,X$, and choose a neighborhood U of x in X that is evenly covered by p. Then $U \cap Bd\,X$ is a neighborhood of x in $Bd\,X$, and $p^{-1}(U \cap Bd\,X) = p^{-1}(U) \cap Bd\,M$. Say $p^{-1}(U) = \bigoplus_\alpha V_\alpha$, where the V_α are open in M and $p: V_\alpha \to U$ is a homeomorphism for each α. (Direct summation denotes disjoint union.) Then $p^{-1}(U \cap Bd\,X) = \bigoplus_\alpha (V_\alpha \cap Bd\,M)$; each set $V_\alpha \cap Bd\,M$ is open in $Bd\,M$, and is carried to $U \cap Bd\,X$ by $Bd\,p$; and $Bd\,p$ restricted to $V_\alpha \cap Bd\,M$ is an embedding, because it is the restriction of the embedding $p|_{V_\alpha}$ to a closed subset of V_α. Therefore $U \cap Bd\,X$ is evenly covered by $Bd\,p$. □

Manifolds have many wonderful properties. We shall have occasion to use only a few. The next two results are well known.

Lemma 2d.4. *Let M be a connected manifold. For every pair of points x and y in $M - Bd\,M$, there is a homeomorphism $h: M \to M$ such that $h(x) = y$, $h|_{Bd\,M} = id_{Bd\,M}$, and $h \simeq id_M$ rel $Bd\,M$.* □

Proposition 2d.5. *Every connected manifold has a simply connected cover.*

Proof. In view of Theorem 2b.6, it suffices to show that every manifold is locally path-connected and semilocally simply connected. Let X be an m-manifold, and let U be a neighborhood of an arbitrary point $x \in X$. We find a simply connected neighborhood of x within U, which will show that X has a basis of simply connected sets. Consequently X locally simply connected, and hence both locally path-connected and semilocally simply connected.

Let h be a homeomorphism of a neighborhood V of x with an open subset of H^m. Then $h(U \cap V)$ is open in H^m, so choose within this set an open ball B around $h(x)$. Since B is convex, it is simply connected, and $h^{-1}(B)$ is homeomorphic to B. Therefore $h^{-1}(B)$ is also simply connected, because path-connectivity and fundamental groups are topological invariants. Furthermore, $h^{-1}(B)$ is open in $U \cap V$, and hence in X. Therefore $h^{-1}(B)$ is a simply connected neighborhood of x, contained within U. □

The next two lemmas are my own inventions, and although they rely on more advanced topics in algebraic topology, their proofs follow directly from standard results.

Lemma 2d.6. *Let M be simply connected, and let U be a path-connected neighborhood of a closed subset $X \subseteq M$. Then each path component of $M - X$ contains exactly one path component of $U - X$.*

Proof. (For those who know singular homology theory.) If $X = U$, then X is both open and closed in M, whence either $X = M$ or $X = \emptyset$ by the connectivity of M. In either case the lemma is trivial. Hence we assume $X \subset U$, and choose a point u of $U - X$. The couple $\{M - X, U\}$ is excisive because $M - X$ and U are open sets that cover M. Hence there is a relative Mayer-Vietoris sequence

$$\cdots \to H_1(M, u) \to H_0(U - X, u) \to H_0(M - X, u) \oplus H_0(U, u) \to H_0(M, u).$$

Because U and M are path-connected, the groups $H_0(M, u)$ and $H_0(U, u)$ are trivial. Furthermore, $H_1(M, u) \approx H_1(M)$ is trivial because M is simply connected. Hence the sequence above takes the form

$$0 \to H_0(U - X, u) \xrightarrow{i_*} H_0(M - X, u) \to 0.$$

Thus the map i_*, which is induced by the inclusion $i\colon (U - X, u) \to (M - X, u)$, is an isomorphism. The groups $H_0(U - X, u)$ and $H_0(M - X, u)$ are free abelian, generated by the path components of $U - X$ and $M - X$, respectively, that do not contain u. For each path component C of $U - X$ with $u \notin C$, there is a path component D of $M - X$ that contains C, and i_* maps the generator of $H_0(U - X, u)$ corresponding to C into the generator of $H_0(M - X, u)$ corresponding to D. For i_* to be an isomorphism means that D does not contain u, and no two path components of $U - X$ are carried by i into the same path component of $M - X$. Hence every path component of $U - X$, including the one that contains u, lies in a unique path component of $M - X$. □

Lemma 2d.7. *Let M be a simply connected, noncompact 2-manifold, and let U be a neighborhood of $x \in M - Bd\,M$ that is homeomorphic to an open ball in R^2. Then every essential loop in $U - x$ is essential in $M - x$.*

Proof. (Uses singular homology and a little homotopy theory.) The set $M - x$ is open because manifolds are Hausdorff, so $\{M - x, U\}$ is an excisive couple in M. Hence there is a Mayer-Vietoris sequence

$$\cdots \to H_2(M) \to H_1(U - x) \to H_1(M - x) \oplus H_1(U) \to H_1(M) \to \cdots.$$

Because M is simply connected, $H_1(M)$ is trivial, and $H_1(U)$ is zero because U is homeomorphic to a contractible space. We can also infer that $H_2(M) = 0$ from the theorem [53] that every connected, noncompact n-manifold satisfies $H_n(M) = 0$. (This theorem is usually stated for manifolds without boundary, but it can be extended to manifolds with boundary as follows. Let M be a connected, noncompact n-manifold with boundary, and let N be the space obtained from the disjoint union $M \oplus (Bd\,M \times [0,1))$ by identifying p with $(p, 0)$ for all $p \in Bd\,M$. Then N is a connected n-manifold without boundary, and N cannot be compact because M is a closed subspace of N that fails to be compact. Hence $H_n(N) = 0$, and $H_n(M) \approx H_n(N)$ because M is a deformation retract of N.) Hence if $i\colon (U - x) \to (M - x)$ denotes the inclusion, then the Mayer-Vietoris sequence takes the form

$$0 \to H_1(U - x) \xrightarrow{i_*} H_1(M - x) \to 0.$$

We conclude that i_* is an isomorphism.

From this we can show that the inclusion i induces a monomorphism of fundamental groups. For any base point y in a space Y, the function that sends a loop $\alpha\colon I \to Y$ at y to the singular 1-cycle α (identifying I with the standard 1-simplex) induces an epimorphism

$$\phi\colon \pi_1(Y, y) \longrightarrow H_1(Y).$$

The kernel of this homomorphism is the commutator subgroup of $\pi_1(Y, y)$, which vanishes if $\pi_1(Y, y)$ is abelian. Let y be any point of $U - x$. The diagram

$$\begin{tikzcd}
\pi_1(U - x, y) \arrow[r, "\phi"] \arrow[d, "i_*"] & H_1(U - x) \arrow[d, "i_*"] \\
\pi_1(M - x, y) \arrow[r, "\phi"] & H_1(M - x)
\end{tikzcd}$$

commutes, as one may easily verify. Furthermore, $U - x$ is homeomorphic to an open ball of R^2 with one point removed. It follows that $U - x$ has the homotopy type of a circle, whence $\pi_1(U - x, y) \approx Z$ is abelian. Therefore the top map in the diagram is an isomorphism. Hence $i_* \circ \phi = \phi \circ i_*$ is an isomorphism, which makes i_* a monomorphism.

To complete the proof, suppose that α is an essential loop in $U - x$, and let y be $\alpha(0)$. Then $[\alpha]_P \neq 0$ in $\pi_1(U - x, y)$, and hence $i_*([\alpha]_P) \neq 0$ in $\pi_1(M - x, y)$. But $i_*([\alpha]_P) = [i \circ \alpha]_P$, so $i \circ \alpha$ is essential in $M - x$. □

Chapter 3

The Topology of Blankets

The main tool in my analysis of single-layer wire routing is the lifting of cuts and wires to a simply connected covering space of the routing region. For this purpose the sketch model is not adequate because cuts and traces in a sketch have their endpoints outside the routing region, and hence cannot be lifted. To avoid this difficulty I retreat to a cleaner model, called the *design model* in which all entities of interest lie wholly within the routing region. The routing region in the design model is a 2-manifold with boundary called a *sheet*, and its simply connected covering space is called a *blanket*. The cuts and wires in the design model are paths called *links* that begin and end on the boundary of the sheet. Since the boundary is part of the sheet, they can always be lifted to the blanket.

The present chapter studies the topological properties of the elements of the design model. (I describe the design model itself at the beginning of Chapter 4, and do not take up the sketch model again until Chapter 8.) Its principal goal is to recapturing some of the simplicity of routing in channels. For example, every cut in a channel divides the channel into two pieces, and one can determine whether a wire is forced to cross the cut by checking whether its endpoints fall on opposite sides of the cut. A cut in a sheet does not separate the sheet, but we prove that every lifting of that cut to the blanket separates the blanket. This fact leads to a good definition of a *necessary* crossing of a cut by a wire and of the *flow* across a cut; see Definition 4b.2.

Looking further at separation properties, we consider how collections of cut liftings separate the blanket. There are two important results in this direction. One says that if a collection of cut liftings in a blanket forms a loop, then that loop has an inside and an outside, and no part of the blanket's boundary lies inside the loop. Consequently, a wire lifting, which must begin and end on the boundary, cannot cross into the loop without also crossing out of it. We use this fact in Chapter 4 to relate the flows across cuts. A second result says that if two cuts are homotopic as links (a concept we will define shortly), then one can choose homotopic liftings of those cuts, and they separate the components of the blanket's boundary in the same way. This fact leads to Proposition 4b.3, which states that homotopic cuts have equal flow.

The chapter concludes with a look at the analogues of rubber bands in the design model. First we show that every path in R^n can be reparameterized to make it canonical without affecting its image, path class, or arc length. Then we prove that every path class of paths in a sheet contains a unique minimum-length canonical path. Such minimum-length paths, called *elastic chains*, will be used for several purposes later on.

Sheets and blankets

Let us begin by defining the elements of the design model. The routing region is a subspace of the plane called a **sheet**: a compact, connected 2-manifold whose boundary consists of two or more disjoint polygons. To make a sheet, start with a polygon P_0, and remove from $P_0 \cup \mathit{inside}(P_0)$ the insides of finitely many disjoint polygons $P_1, \ldots, P_n$ that lie inside P_0. If $n \geq 1$, the resulting space is a sheet whose boundary has connected components $P_0, P_1, \ldots, P_n$. These subspaces are the **fringes** of the sheet. The insides of the polygons $P_1, \ldots, P_n$ and the outside of P_0 form the routing obstacles. Because sheets are connected manifolds, Proposition 2d.5 shows that every sheet has a simply connected cover, which we call a **blanket**. And since sheets are connected and locally path-connected, Theorem 2b.7 shows that all blankets of a sheet are equivalent. Hence we can speak of "the" blanket of a sheet. By Proposition 2d.3, every blanket is a 2-manifold with boundary.

The simplest sort of blanket is depicted in Figure 2b-1, if one takes the borders of the annulus to be polygons. In this example the sheet has only two fringes. The blanket for a sheet with 3 or more fringes is harder to visualize and to draw, though I have made an attempt in Figure 2b-2. If one is concerned with the covering map, then one should envision the blanket as infinitely many layers lying above the sheet, connected so as to satisfy the following condition: a path in the blanket is a loop if and only if its projection to the sheet is an inessential loop. If one is concerned only with the intrinsic properties of the blanket, however, then the representation of Figure 3-1 is helpful; it embeds the blanket in a bounded region of the plane.

Our primary objects of interest are paths of various kinds. We study paths in a sheet by lifting them to the sheet's blanket. (By Theorem 2b.3, the Lifting Theorem, paths can always be lifted.) "Lifting" will always mean lifting from the sheet to its blanket. For instance, if Φ is a set of paths in a sheet, then a **Φ-lifting** is any path in the sheet's blanket whose projection to the sheet is a member of Φ.

Flat manifolds

Sheets, blankets, and all their submanifolds have a very special property: they are **flat**. A flat m-manifold M is one that comes equipped with a local embedding

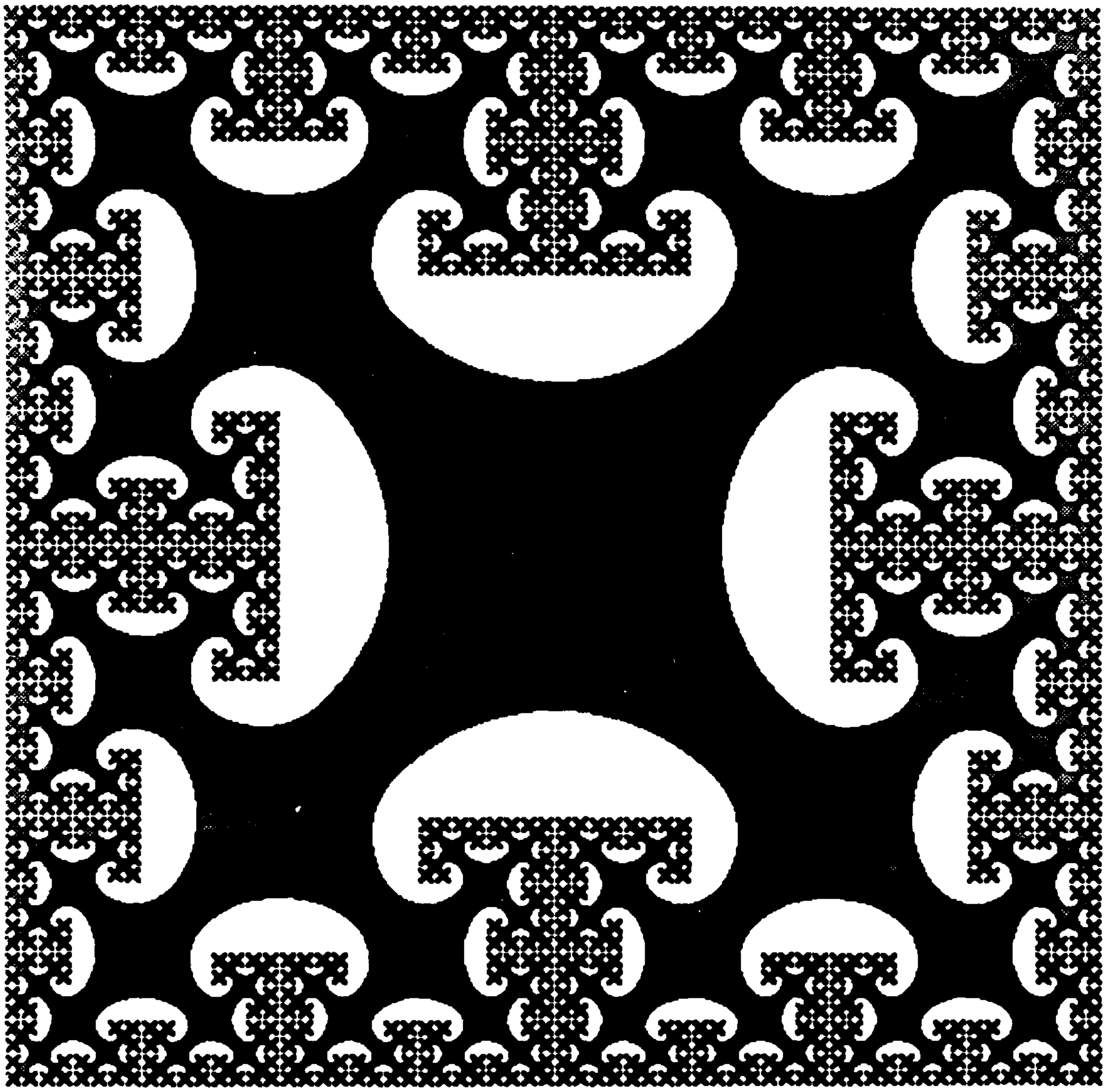


Figure 3-1. *One way to visualize a blanket.* This figure forms the basis for many subsequent pictures of blankets. (Later figures show only part of the blanket and parts of a few fringes.) All blankets are homeomorphic either to $R \times I$ (as in Figure 2b-1) or to the shaded subspace of the plane. Specifically, every sheet with 3 or more fringes has this space as a blanket, though the covering map varies. Part of this surface is shown in Figure 2b-2.

$h: M \to R^m$. Any sheet S is flat, because it comes with an inclusion $i: S \to R^2$. (Inclusions are always embeddings, and hence local embeddings as well.) Every blanket is flat, for if M is a blanket with covering map $p: M \to S$, then p is a local homeomorphism, and hence $i \circ p: M \to R^2$ is a local embedding. A submanifold N of a flat manifold M is naturally a flat manifold, for if $h: M \to R^m$ is a local embedding, so is $h|_N$. Of particular importance to us are the **scraps** of a blanket: its simply connected, open submanifolds. All scraps are flat manifolds.

Flat manifolds inherit many nice properties of Euclidean space, such as a notion of linearity for paths. Let M be flat, with local embedding $h: M \to R^m$. A path α in M is **linear** if $h \circ \alpha$ is linear, **straight** if $h \circ \alpha$ is linear and nonconstant, and **bent** if there is a point $t \in I$ such that $\alpha_{0:t}$ and $\alpha_{1:t}$ are straight and intersect at $\alpha(t)$ alone. More generally, α is **piecewise linear**, abbreviated **PL**, if $h \circ \alpha$ is piecewise linear, and **piecewise straight** if $h \circ \alpha$ is piecewise linear and none of its segments is constant. A **simple path** is piecewise linear and injective; a **simple loop** is the same except that its endpoints coincide. Straight and bent paths are always simple. If x is a point in I, we say α is **linear at x**, **straight at x** or **bent at x** if there is an interval $[s,t]$ containing a neighborhood of x such that $\alpha_{s:t}$ is linear, straight, or bent, respectively. All of these properties except simplicity are preserved by lifting from a sheet to its blanket and by projecting from a blanket to its sheet. For example, if α is a lift of a path β, then α is straight at x if and only if β is straight at x. If β is simple then so is α, but the converse is false.

Another notion that flat manifolds inherit is that of arc length. Given a norm $|\cdot|$ on R^m, one can define the arc length of a path α in M as $|h \circ \alpha|$. If M is path-connected, one thereby obtains a metric for M: define the distance between two points in M to be the infimum of the arc lengths of all paths between those points. You may check that this distance function is a topological metric on M.

Links and link homotopy

The paths in a manifold fall into different categories depending on where they touch the boundary of the manifold. A path α in a manifold M is a **link** if $\alpha^{-1}(Bd\, M) = \{0,1\}$, a **half-link** if $\alpha^{-1}(Bd\, M) = \{0\}$, and a **mid-link** if $\alpha^{-1}(Bd\, M)$ is empty. Links, half-links, the reverses of half-links, and mid-links are collectively called **sublinks**. A **chain** for a path α, so called because it may contain one or more links, is any path in $[\alpha]_P$. For any manifold M, we call the components of $Bd\, M$ the **fringes** of M. The fringes that contain the endpoints of a sublink are the **terminals** of the sublink. A link has either one or two terminals, a half-link has one, and a mid-link has none.

The notion of homotopy for links is very important because it applies to all cuts and wires. Two links α and β in a manifold M are **link-homotopic**, written $\alpha \simeq_L \beta$, if there is a homotopy $H: I \times I \to M$ between α and β such that $H(\{0,1\} \times I) \subseteq Bd\, M$. In other words, as α is deformed into β, its endpoints must stay on their respective fringes. Thus link-homotopic links have the same terminals. The map H is called a **link homotopy**. One may check that the relation of being link-homotopic (also called link homotopy) is an equivalence relation; the set of links that are link-homotopic to α is denoted $[\alpha]_L$. Two links that are path-homotopic are also link-homotopic, so the path-homotopy class $[\alpha]_P$ is always a subset of $[\alpha]_L$.

Liftings of links

Because the majority of the lemmas and propositions in the next four chapters involve lifting paths from a sheet to its blanket, some further remarks about lifting are in order. Suppose the blanket M covers the sheet S via the map p. By Lemma 2d.3, the boundary of the blanket lies over the boundary of the sheet. Hence a lifting of a link is a link, a lifting of a half-link is a half-link, and a lifting of a mid-link is a mid-link. An elementary but important fact is that the liftings of a simple path are disjoint. For let α and β lift the simple path γ, and suppose $\alpha(s) = \beta(t)$. Then $\gamma(s) = \gamma(t)$, whence $s = t$. Hence by uniqueness of liftings (Theorem 2b.2) we have $\alpha = \beta$.

Another useful fact is that all the lifts of a path in S have the same topological properties: if α and β lift the same path, then there is a covering transformation $T: M \to M$ such that $T \circ \alpha = \beta$. For by Proposition 2b.7, the covering spaces $(M, \alpha(0))$ and $(M, \beta(0))$ are equivalent; there is a covering transformation $T: M \to M$ that carries $\alpha(0)$ to $\beta(0)$. So $T \circ \alpha$ lifts the same path as α, and it agrees with β at 0; hence $T \circ \alpha = \beta$ by uniqueness of liftings. Moreover, T is a homeomorphism, and therefore α and β are topologically indistinguishable.

3A. Constructing Paths in Blankets

One drawback of working with blankets is that their geometry and topology are unfamiliar. Whereas in the plane one can take for granted many theorems of Euclidean geometry and geometric topology, the analogous facts about blankets are far less intuitive. Hence the need for the present chapter, which collects basic results about blankets.

We begin with several methods for constructing paths and links in blankets. First we show there exists a simple link, half-link, or mid-link between every two distinct points in a blanket. Then we characterize link homotopy in terms of path homotopy, and we prove that two links in a blanket are link-homotopic if and only if they begin and end on the same fringes. Most importantly, we relate link homotopy in a sheet to link homotopy in its blanket. Homotopic links in a blanket, when projected to the sheet, remain homotopic; homotopic links in a sheet can always be lifted so that their liftings are homotopic.

Existence of simple paths

Because blankets are connected manifolds, they are path-connected. That is, for every two points in a blanket, there is a path that connects them. Since I work only with piecewise linear paths, I need to know that the path can always be made piecewise linear. We can prove something stronger: the path can always be made simple, and its middle need never intersect a fringe. Two lemmas are helpful in proving this claim. The first says that one can remove all self-intersections from a piecewise linear path.

Lemma 3a.1. *For any PL path α in a flat manifold, there is a simple path β in $Im\,\alpha$ with the same endpoints as α, and $\|\beta\| \le \|\alpha\|$ in any norm $\|\cdot\|$.* □

The proof of this lemma is an induction on the number of pairs of segments of α that intersect. I omit the details.

The second lemma states that one cannot disconnect a manifold by removing all or part of its boundary.

Lemma 3a.2. *If M is a connected manifold and $X \subseteq Bd\,M$, then $M - X$ is connected.*

Proof. Let N denote $M - X$, and suppose that N is not connected. Then there are nonempty open sets U and V that partition N. Let $Cl\,U$ and $Cl\,V$ denote the closures of U and V in M. Because M is connected, $Cl\,U$ and $Cl\,V$ must intersect, or else their complements, which are nonempty open sets, would partition M. Let x be a point of $(Cl\,U) \cap (Cl\,V)$; it cannot lie in N, and hence must lie in X. Take a boundary patch $h: W \to H^n$ around x whose image is an open ball. Then $h(W \cap N)$ is the connected set $h(W) - Bd\,H^n$ with perhaps some points of closure added, so $h(W \cap N)$ is also connected. Hence $W \cap N$ is connected since h is a homeomorphism. But $W \cap N$ is the union of its disjoint open subsets $W \cap U$ and $W \cap V$, neither of which is empty, because x is a point of closure of both U and V. Thus $W \cap N$ is not connected, a contradiction. □

Armed with Lemmas 3a.1 and 3a.2, we show that for any two points in a blanket, there is a simple link, half-link, or mid-link connecting them.

Proposition 3a.3. *Every pair of points in a scrap M can be connected by a simple path whose middle lies in $M - Bd\,M$.*

Proof. Set N equal to $M - Bd\,M$. We first prove the proposition in the case where both points lie in N. Being an open subset of a flat manifold, N itself is a flat manifold. The preceding lemma shows that N is connected. Let $p: M \to R^m$ be the local embedding associated with M. Say that an open set $V \subseteq N$ is **nice** if h maps V homeomorphically onto a convex subset of R^m. Every point of N has a nice neighborhood. Define an equivalence relation $\sim$ on the points of N by setting $x \sim y$ if there is a finite sequence of nice sets $V_1, \ldots, V_n$ such that:

(1) $x \in V_1$ and $y \in V_n$; and

(2) V_i meets V_{i+1} whenever $1 \leq i < n$.

The equivalence classes of $\sim$ are open, and form a partition of N; since N is connected, there can be only one equivalence class.

So for any two points $x, y \in N$, there is a finite sequence of nice sets $V_1, \ldots, V_n$ satisfying (1) and (2) above. We use this sequence to construct a simple path in M from x to y. Choose points $x = x_0, x_1, \ldots, x_n = y$ such that $x_i \in V_i \cap V_{i+1}$ for all i satisfying $1 \leq i < n$. Because each set $p(V_i)$ is convex, the linear path λ_i from $p(x_{i-1})$ to $p(x_i)$ lies in $p(V_i)$ for each i. Let α_i be $(p|_{V_i})^{-1} \circ \lambda_i$, and let α be the concatenated path $\alpha_1 \star \cdots \star \alpha_n$. Then α is piecewise linear, and runs from x to y. Lemma 3a.1 reduces α to a simple path from x to y.

To complete the proof, suppose that one of the points to be connected, say x, lies on *Bd M*. There is a patch $h \circ p$, defined on a neighborhood U of x, such that $p(U)$ is polygonal. Take any straight path from $p(x)$ whose middle lies in $Int\, p(U)$, and lift it to a path α in U starting at x. Then α is a straight half-link in M. The previous lemma proves the existence of a simple path γ in N from $\alpha(1)$ to y, and the concatenated path $\alpha \star \gamma$ is a PL half-link in M from x to y. Lemma 3a.1 reduces this path to a simple half-link from x to y. The same technique handles the case in which both x and y lie on *Bd M*. □

Link homotopy

One can characterize link homotopy in terms of path homotopy, as the next lemma shows.

Lemma 3a.4. *Two links α and β in a manifold M are link-homotopic if and only if there exist paths κ and ν in Bd M such that $\alpha \star \kappa \star \widehat{\beta} \star \widehat{\nu}$ is an inessential loop in M.*

Proof. This is a consequence of Lemma 2a.9. For there to be a link homotopy between α and β means that there is a map $f: Fr(I \times I) \to M$ with an extension F over $I \times I$ such that $f(\cdot, 0) = \alpha$, $f(\cdot, 1) = \beta$, and the paths $\nu = f(0, \cdot)$ and $\kappa = f(1, \cdot)$ run in *Bd M*. By Lemma 2a.9, the existence of the extension F is equivalent to $f \circ \delta$ being inessential, where δ is the loop

$$\delta = (\cdot, 0) \star (1, \cdot) \star \widehat{(\cdot, 1)} \star \widehat{(0, \cdot)}: I \to I \times I.$$

But $f \circ \delta$ is just $\alpha \star \kappa \star \widehat{\beta} \star \widehat{\nu}$, so the proof is complete. □

In a simply connected manifold the condition that the loop be inessential is redundant. Since fringes are path-connected, we obtain the following important corollary.

Corollary 3a.5. *Two links in a blanket are link-homotopic if and only if they have the same terminals.* □

Here we use the convention that α and β **have the same terminals** if $\alpha(0)$ lies on the same fringe as $\beta(0)$, and $\alpha(1)$ lies on the same fringe as $\beta(1)$.

If link-homotopic links in a blanket are projected to the sheet, they remain link-homotopic. For if F is a link homotopy between α and β, and if $p: M \to S$ is the covering map, then $p \circ F$ is a link homotopy between $p \circ \alpha$ and $p \circ \beta$. The next lemma is a partial converse: given link-homotopic links in a sheet, we can lift them to obtain link-homotopic links in the blanket.

Proposition 3a.6. *Let α and β be link-homotopic links in a sheet S, and let M be a blanket of S. There is a bijective correspondence between the lifts of α to M and the lifts of β to M, and corresponding lifts are link-homotopic.*

Proof. Let $p: M \to S$ be the covering map. Choose a link homotopy $F: I \times I \to S$ between α and β, and let $\widetilde{\alpha}$ be any lift of α. We say that $\widetilde{\alpha}$ corresponds to a lift $\widetilde{\beta}$ of β if there is a link homotopy between $\widetilde{\alpha}$ and $\widetilde{\beta}$ that lifts F.

By symmetry, it suffices to show that to each lift $\widetilde{\alpha}$ of α there corresponds a unique lift $\widetilde{\beta}$ of β. Let $\widetilde{\alpha}$ be given. Because $I \times I$ is a convex subset of R^2, it is locally path-connected and simply connected, and hence by the Lifting Theorem (2b.3), F has a lift $\widetilde{F}: I \times I \to M$ such that $\widetilde{F}(0,0) = \widetilde{\alpha}(0)$. Theorem 2b.2 shows this lift to be unique, so there is only one choice for $\widetilde{\beta}$, namely $\widetilde{\beta} = \widetilde{F}(\cdot, 1)$. We see that $\widetilde{\beta}$ is a lift of β, because

$$p \circ \widetilde{\beta} = p \circ \widetilde{F}(\cdot, 1) = F(\cdot, 1) = \beta.$$

I claim that $\widetilde{F}$ is a link homotopy between $\widetilde{\alpha}$ and $\widetilde{\beta}$. Two things must be shown: that $\widetilde{F}(\cdot, 0) = \widetilde{\alpha}$, and that $\widetilde{F}(\{0,1\} \times I)$ is contained in $Bd\,M$. The second is easy. Because F is a link homotopy, we have

$$p \circ \widetilde{F}(\{0,1\} \times I) = F(\{0,1\} \times I) \subseteq Bd\,S.$$

Now $p^{-1}(Bd\,S) = Bd\,M$ by Lemma 2d.3, and hence $\widetilde{F}(\{0,1\} \times I) \subseteq Bd\,M$. To show that $\widetilde{F}(\cdot, 0)$ and $\widetilde{\alpha}$ coincide, note that they are lifts of α that agree at one point (namely 0), and apply uniqueness of liftings (Theorem 2b.2). □

Lifting of convergent sequences

The last result in this section concerns the lifting of another relation among paths: uniform convergence. Given a uniformly convergent sequence of paths in the sheet, we can lift them to the blanket so that the limit of the lifts is a lift of the limit.

Lemma 3a.7. *Let M be a blanket of a sheet S. Let $\langle \alpha_n \rangle$ be a sequence of paths in S that converges uniformly to a path α, and let β be a lift of α to M. There is a sequence of paths $\langle \beta_n \rangle$ that converges uniformly to β, and β_n lifts α_n for each n.*

Proof. Let $p\colon M \to S$ be the covering map. Choose ϵ smaller than the minimum distance between fringes of S, and small enough that whenever two points on a fringe V of S are separated by a distance ϵ or less, they lie on adjacent segments of the polygon V. Let $P \subset S \times S$ be the set $\{(p,q) : |p-q| < \epsilon\}$, and define a function $L\colon P \times I \to S$ as follows. The path $L(p,q,\cdot)$ is the linear path from p to q if this path lies in S. Otherwise, let V be the unique fringe of S that $p \triangleright q$ intersects. Because V is a convex polygon, $p \triangleright q$ crosses exactly two segments of V. These segments are adjacent. Let v be their common vertex, and define $L(p,q,\cdot)$ to be the path $(p \triangleright v) \star (v \triangleright q)$, parameterized according to arc length. Then L is a continuous function on $P \times I$. In addition, there is a constant K such that the arc length of $L(p,q,\cdot)$ is at most $K\,|p-q|$.

We construct the sequence $\langle \beta_n \rangle$ as follows. Let ϵ_n be $\sup_{t \in I} |\alpha_n(t) - \alpha(t)|$; we have $\epsilon_n \to 0$ as $n \to \infty$. When $\epsilon_n \geq \epsilon$, let β_n be an arbitrary lift of α_n. Otherwise, let F_n be the homotopy between α_n and α given by

$$F_n(s,t) = L(\alpha_n(s), \alpha(s), t).$$

Because L is continuous, so is F_n. Let G_n be a lift of F_n satisfying $G_n(\cdot,1) = \beta$, and set $\beta_n = G_n(\cdot,0)$. Then β_n lifts α_n, and the distance between β_n and β is

$$\begin{aligned}\sup_{t \in I} \inf\{\, |p \circ \sigma| : \sigma\colon \beta_n(t) \rightsquigarrow \beta(t) \,\} &\leq \sup_{t \in I} |p \circ G_n(t,\cdot)| \\ &= \sup_{t \in I} |L(\alpha_n(t), \alpha(t), \cdot)|\,.\end{aligned}$$

Since the distance between $\alpha_n(t)$ and $\alpha(t)$ is at most ϵ_n, this quantity is bounded by $K\epsilon_n$. Therefore the paths $\langle \beta_n \rangle$ converge uniformly to β. □

3B. Separation Results

In this section we consider more of the global topological properties of blankets. The main result, which is fundamental to my entire approach to wire routing, is that every simple link in a blanket splits it into two scraps. We build on this result to show that any simple loop of k links splits a blanket into $k + 1$ scraps, one of which contains no fringes. If these properties seem obvious in view of Figure 3-1, you may consider this section as providing formal evidence that Figure 3-1 is an accurate representation of a blanket.

The topology of fringes

Every fringe of a sheet is a polygon, and hence homeomorphic to the circle S^1. It should come as no surprise, therefore, that every fringe of a blanket is homeomorphic to the real line R^1. To prove this fact we first need one lemma.

Lemma 3b.1. *Every fringe of a sheet is a retract of the sheet.*

Proof. Let F be a fringe of the sheet S. Suppose first that S lies outside F, by which I mean $S \subset F \cup \mathit{outside}(F)$. By Lemma 2c.3 there is a retraction of $F \cup \mathit{outside}(F)$ onto F, which when restricted to S gives a retraction of S onto F.

The other possibility is that S lies inside F. Because S has two or more fringes, there is a point x in $\mathit{inside}(F) - S$. Let the map h be inversion with respect to the unit circle centered at x. Since h is its own inverse, it is a homeomorphism of $R^2 - x$ with itself. Now $h(S)$ is a sheet that lies outside the fringe $h(F)$. Hence there is a retraction r from $h(S)$ onto $h(F)$, and the map $h \circ r \circ h$ is a retraction of S onto F. □

Let F be a fringe of a sheet S. As shown in Section 2A, the fact that F is a retract of S implies that the inclusion $i: F \rightarrow S$ induces a monomorphism of fundamental groups: every essential path in F is essential in S. We use this fact in the following lemma and elsewhere.

Lemma 3b.2. *Every fringe of a blanket is homeomorphic to R^1.*

Proof. Let A be a fringe of the blanket M, and let $p: M \rightarrow S$ be the covering map. By Lemma 2d.3, the fringe A covers a fringe F of S via the map $p|_A$. We show that A is simply connected, and thence Proposition 2b.7 shows that A is homeomorphic to any other simply connected covering space of F. Since F is homeomorphic to the circle S^1, the real line R^1 is one such covering space.

Because $Bd\,M$ is a manifold, its component A is a connected manifold and hence path-connected. It remains to show that every loop α in A is inessential in A. Certainly α is inessential in M, because M is simply connected. Hence $p \circ \alpha$ is inessential in S. Because F is a retract of S, by Lemma 3b.1, the loop $p \circ \alpha$ is inessential in F. But $p|_A: A \to F$ is a covering map, so any lift of $p \circ \alpha$ to A is inessential in A. In particular, α is inessential in A. □

Neighborhoods of sublinks

To determine how a set X separates a blanket, we apply Proposition 2d.6 to a neighborhood U of X whose properties we know. Here X is the image of a simple path in a flat 2-manifold. The neighborhoods we use are called *tubular* because they look like thin tubes about the simple path in question. A tubular neighborhood of X has no holes: it separates the manifold essentially as X does.

Definition 3b.3. Let α be a simple sublink in a flat 2-manifold M. A neighborhood N of $Im\,\alpha$ is **tubular** if there is a piecewise linear homeomorphism $h: I \times I \to Cl\,N$ whose inverse k has the properties shown in table 3b-1.

$\alpha^{-1}(Bd\,M)$	$k(Fr\,N)$	$k(Bd\,M)$	$k \circ \alpha$
$\emptyset$	$\bigcup e_i$	$\emptyset$	$p_1 \triangleright p_2$
$\{0\}$ or $\{1\}$	$e_1 \cup e_3 \cup e_4$	e_2	$p_0 \triangleright p_2$
$\{0, 1\}$	$e_1 \cup e_4$	$e_2 \cup e_3$	$p_0 \triangleright p_3$

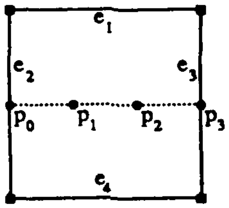


Table 3b-1. *Requirements for a tubular neighborhood.* For N to be a tubular neighborhood of $Im\,\alpha$, the homeomorphism k must carry the path α and the sets $Fr\,N$ and $Cl\,N \cap Bd\,M$ onto certain parts of $I \times I$, which differ depending on whether α is a mid-link, half-link, or link. The points $p_0, \ldots, p_3$ are given by $p_i = (\frac{i}{3}, \frac{1}{2})$, while the line segments $e_1, \ldots, e_4$ are $e_1 = I \times 1$, $e_2 = 0 \times I$, $e_3 = 1 \times I$, and $e_4 = I \times 0$.

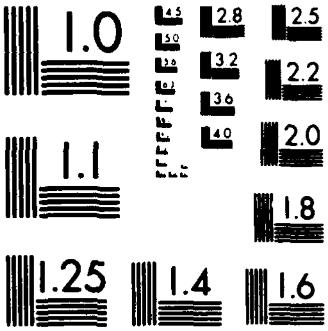
Of course, we need to know that every simple sublink has tubular neighborhoods. To prove this rigorously would be very tedious. The following lemma shows only how to construct the neighborhoods; Figure 3b-2 suggests how one might prove that they are, in fact, tubular.

Lemma 3b.4. *Let α be a simple sublink in scrap. Every neighborhood of $Im\,\alpha$ contains a tubular neighborhood of $Im\,\alpha$.*

Proof. Let M be a scrap of the blanket B, and let $p: B \to S$ be the covering map. Write α as the concatenation of finitely many paths α_i such that $p \circ \alpha_i$ is a line segment for each i. By subdividing these line segments if necessary, we may assume that for each i, the image of α_i sits inside a neighborhood V_i such that $p|_{V_i}$ is an embedding. Choose a positive number ϵ smaller than the following quantities:

AD-A186 990 SINGLE-LAYER WIRE ROUTING(U) MASSACHUSETTS INST OF TECH 2/4
CAMBRIDGE LAB FOR COMPUTER SCIENCE F M MALEY AUG 87
MIT/LCS/TR-403 N00014-80-C-0622

UNCLASSIFIED F/G 9/1 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

(1) The minimum distance between a path α_i and the complement $B - V_i$ of the corresponding neighborhood.

(2) The minimum distance between paths α_i and α_j, over all i and j with $|i - j| > 1$.

(3) The minimum distance between the compact set $Im\,\alpha$ and the fringe edges of B that do not intersect $Im\,\alpha$.

(4) The distance from $Im\,\alpha$ to the closed set $B - M$, if the latter is nonempty.

Let N be the set of points whose distance from $Im\,\alpha$ is less than $\epsilon/2$. Then N has the desired properties; see Figure 3b-2. □

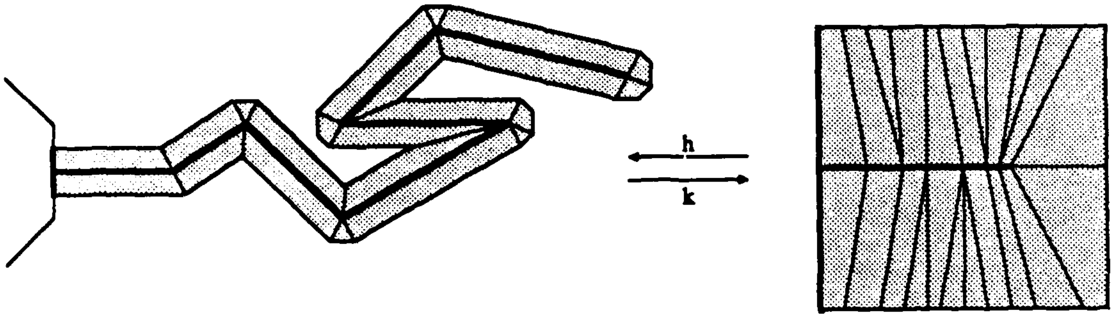


Figure 3b-2. *Construction of a tubular neighborhood.* If α is a simple sublink in a scrap, then for sufficiently small ϵ the set of points N whose distance from $Im\,\alpha$ in the norm $\|\cdot\|$ is less than ϵ is a tubular neighborhood of $Im\,\alpha$. One can set up a homeomorphism between $Cl\,N$ and $I \times I$ (the map k with inverse h, as in Definition 3b.3) that takes triangles and quadrilaterals in $Cl\,N$ to triangles and quadrilaterals in $I \times I$, and has the properties listed in Table 3b-1.

Threads and half-threads

Figure 3-1 suggests that a simple link in a blanket should split it into two parts, and this claim we now prove. To be proper, one should not speak of a path separating a space, but rather of the image of the path doing so. Some new terminology is therefore helpful: a **thread** is the image of a simple link, and a **half-thread** is the image of a simple half-link. We could also consider "mid-threads", but they turn out not to be very useful. To summarize: threads separate scraps, but half-threads (and mid-threads) do not.

Lemma 3b.5. *Removing a half-thread from a scrap leaves a scrap.*

Proof. Let α be a simple half-link in a scrap M, let A denote its image, and let U be a tubular neighborhood of A with homeomorphism $h\colon I \times I \to Cl\,U$. Because A is a compact subset of the Hausdorff space M, it is closed, and hence $M - A$ is open in M. Therefore $M - A$ is an open subspace of a blanket. To show that $M - A$ is simply connected, it suffices in view of Lemma 2a.7 to find a simply connected deformation retract of $M - A$. In the notation of Table 3b-1, one can construct a deformation retraction F of $I \times I - \overline{p_0p_2}$ onto $e_1 \cup e_3 \cup e_4$. Then $h \circ F \circ h^{-1}$ is a deformation retraction of $Cl\, U$ onto $Fr\, U$. Because it fixes $Fr\, U$, this map extends to a deformation retraction of $M - A$ onto $M - U$. In a similar way one can construct a deformation retraction of M onto $M - U$. Since M is simply connected, Lemma 2a.7 shows that $M - U$ is simply connected. □

Proposition 3b.6. *Removing a thread from a scrap leaves two scraps whose common frontier is the thread.*

Proof. The construction is illustrated in Figure 3b-3 below. Let α be a simple link in a scrap M, let C denote its image, and let U be a tubular neighborhood of C with homeomorphism $h: I \times I \to Cl\, U$. From Table 3b-1 we see that h carries $I \times \frac{1}{2}$ and $I \times (0,1)$ onto C and U, respectively. Hence $U - C$ has two path components, call them A' and B', and their closures in U include C. Lemma 2d.6 now implies that $M - C$ has exactly two path components, each containing a path component of $U - C$. Call them A and B, and say $A \supseteq A'$ and $B \supseteq B'$. The set $M - C$ is open in M, because C is a compact subset of the Hausdorff space M, and therefore closed in M. Hence $M - C$ is locally path-connected (because M is), and so its path components A and B are open. Therefore A and B are the components of $M - C$; we have $Cl\, A \subseteq M - B$ and $Cl\, B \subseteq M - A$. But we also know

$$Cl\, A \supseteq Cl\, A' \supset C \qquad \text{and} \qquad Cl\, B \supseteq Cl\, B' \supset C.$$

Together these facts imply $Cl\, A = A \cup C$ and $Cl\, B = B \cup C$, whence $Fr\, A = C = Fr\, B$ because A and B are open.

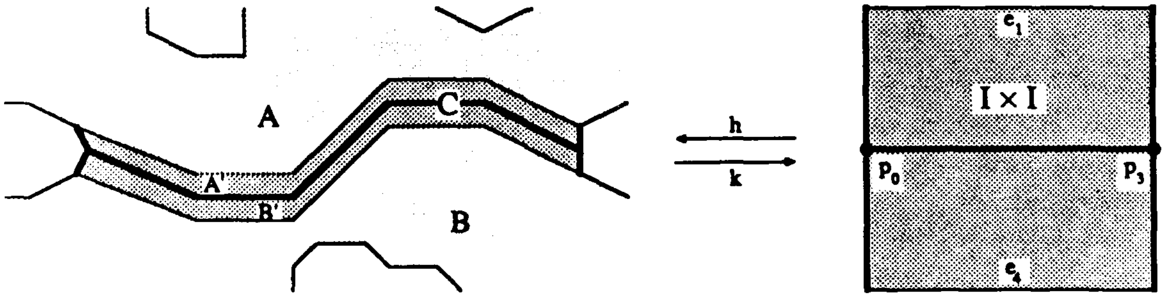


Figure 3b-3. *How a thread separates a scrap.* The tubular neighborhood U of the thread $Im\, \alpha$ shows us how $Im\, \alpha$ is embedded in the blanket. Every topological relationship among the path $p_0 \triangleright p_3$ and the components of its complement in $I \times I$ also obtains for α and the components of its complement in $Cl\, U$.

By the symmetry between A and B, it suffices to prove that A is a scrap. We know that A is open, so we need only show that A is simply connected. The set $I \times I - \overline{p_0p_3}$ has a deformation retraction onto $e_1 \cup e_4$. Pulling back via h, one obtains a deformation retraction of $(Cl\,U) - C$ onto $Fr\,U$. Since this map fixes $Fr\,U$, it extends to a deformation retraction of $M - C$ onto $M - U$. Restricting to A, we obtain a deformation retraction of A onto $A - U$. For similar reasons, there is a deformation retraction of $A \cup C$ onto $A - U$. Hence by Lemma 2a.7, if we show that $A \cup C$ is simply connected, it will follow that $A - U$ and A are simply connected. We exhibit $A \cup C$ as a retract of the simply connected space M. The map α is an embedding of I as a closed subspace C of $B \cup C$, and the latter space is normal. (The blanket containing $B \cup C$ is metrizable, hence $B \cup C$ is metrizable, and thus normal.) Since I is an a absolute retract, there must be a retraction $r\colon B \cup C \rightarrow C$. This map r may be extended over M by making it the identity on A. Then r is a retraction of M onto $A \cup C$. Because M is simply connected, so is the retract $A \cup C$ of M. □

Weaving threads into webs

Building on Proposition 3b.6, one can determine how groups of threads partition a blanket. Simple loops are especially important to analyze. Let λ be a simple loop in a blanket M, and suppose that $Im\,\lambda \cap Bd\,M$ has $k > 0$ components, each containing more than one point. Then λ is called a **loop of k links**. In addition, the set $Cl(Im\,\lambda - Bd\,M)$ is the union of k disjoint threads, and is called a **web of k threads**. A straightforward induction shows that a web of k threads splits a blanket into $k + 1$ parts.

Lemma 3b.7. *Removing a web of k threads from a blanket leaves exactly $k+1$ scraps. One has the entire web as frontier, while the others border on one thread each.*

Proof. Let λ be a loop of k links, and let $Im\,\beta_1, \ldots, Im\,\beta_k$ be the threads contained in $Im\,\lambda$. We apply Proposition 3b.6 to each of the threads $Im\,\beta_i$. First consider $Im\,\beta_1$: it separates the blanket into two scraps, only one of which contains the remaining links $\beta_2, \ldots, \beta_k$, because the loop λ is simple. The thread $Im\,\beta_2$ separates this scrap into two scraps, one of which contains $\beta_3, \ldots, \beta_k$. Continue in this way, obtaining $k + 1$ scraps. At each stage, exactly one of the scraps contains the remaining threads, and borders on all the threads removed; each of the other scraps borders on one thread $Im\,\beta_i$. □

In Lemma 3b.7 the special scrap is called the **inside** of the loop, or of the web. Figure 3-1 suggests strongly that the inside of a web is compact and that it contains only parts of fringes. The following rather technical result bears out these conjectures.

Proposition 3b.8. *No fringe lies inside a web of threads.*

Proof. We begin by showing that the closure of the inside of a web is simply connected. Let $T_1, \ldots, T_k$ be the threads that make up a web W of k threads in a blanket M. Let B denote the inside of W, and for $1 \leq i \leq k$, let A_i be the component of $M - W$ that borders only the thread T_i. For each i, the absolute retract I is embedded in the normal space $A_i \cup T_i$ as the closed set T_i, so there is a retraction r_i of $A_i \cup T_i$ onto T_i. Define a retraction $r\colon M \to B \cup W$ by $r(x) = r_i(x)$ if $x \in A_i \cup T_i$, and $r(x) = x$ if $x \in B \cup W$. These definitions agree on their intersection, which is W, so r is indeed continuous. Its image is the space $C = Cl\, B = B \cup W$. Thus C is a retract of M, and because M is simply connected, so is C.

In the remainder of the proof we prove that $Cl\, B$ is compact, whence it follows that B includes no fringe of M. For if X is a fringe of M, then by Lemma 3b.2, it is homeomorphic to R^1. Hence X contains an infinite discrete subspace Z, and since X is closed in M, this subset is discrete in M. The points of Z cannot all lie in a compact subspace C of M, and so neither does X.

Because C is simply connected and contains $Im\, \lambda$, there is a path homotopy $F\colon I \times I \to C$ between the loop λ and the constant loop at $\lambda(0)$. Suppose that $x \in C - Im\, F$. Certainly $x \notin Bd\, C$, because $Bd\, C \subset Im\, \lambda \subseteq Im\, F$. Let y be an arbitrary point of $B - Bd\, B$. Because $Cl\, B$ is a connected manifold, there is by Lemma 2d.4 a homeomorphism $h\colon C \to C$ that fixes $Bd\, C$ and carries x onto y. Then $h \circ F$ is a path homotopy between λ and a constant loop, and $y \notin Im(h \circ F)$. Hence we may choose any point $x \in C - Bd\, C$ and assume $x \notin Im\, F$.

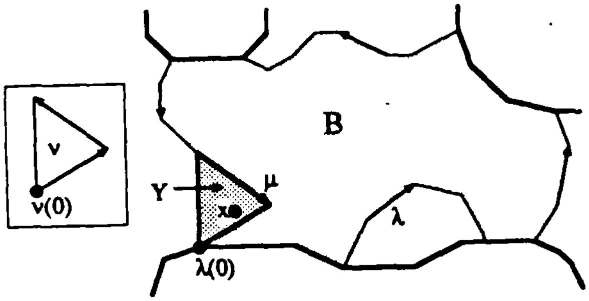


Figure 3b-4. *The inside of a loop of links.* The path λ is a loop of 4 links in the blanket M. It is essential in the space $M - x$, because $\lambda \simeq_P \mu \star \nu$, and ν is essential in $M - x$ but μ is not. In fact, λ is essential in $M - x$ for any point x inside λ.

We choose a point $x \in C - Bd\, C$ very near W. Let U be a neighborhood of $\lambda(0)$ that intersects only two line segments of W. We may assume that $\lambda(0)$ lies in $Bd\, M$. If $p\colon M \to R^2$ denotes the local embedding attached to M, we may also assume that p embeds $U \cap C$ as a polygonal region in R^2. Choose $x \in U \cap B$. As shown in Figure 3b-4, there is a loop ν at $\lambda(0)$ and a loop of links μ at $\lambda(0)$ such that

- $\lambda \simeq_P \mu \star \nu$,
- the loop $p \circ \nu$ is a simple polygon that encloses $p(x)$, and

- x lies outside μ.

Since the closure of the inside of μ is simply connected, μ is inessential in that subspace of $C - x$, and hence μ is inessential in $C - x$. On the other hand, ν is essential in $U \cap C - x$, because $p \circ \nu$ is essential in $p(U \cap C) - p(x)$, and $p|_{U \cap C}$ is an embedding.

Lemma 2d.7 applies to the neighborhood U of M; it says that ν is essential in $M - x$. But μ is not, because x lies outside μ. Therefore $[\lambda]_P = [\mu \star \nu]_P = [\nu]_P \neq 0$ in the fundamental group of $M - x$. But F is a path homotopy in $M - x$ from λ to a constant map. This contradiction shows that our assumption $C \neq Im\, F$ was faulty. So $C = Im\, F$, which is compact because $I \times I$ is compact. □

3C. Properties of Separations

When a loop splits the plane or a blanket, there is a convenient distinction between the inside of the loop and its outside. And when a link separates a blanket, we can distinguish between the left-hand side of the link and the right-hand side. This section explores the implications of the distinctions between left and right, and between inside and outside. One important result is that link-homotopic simple links partition the fringes of a blanket in the same way. Chapter 4 uses this result to show that the necessity of a crossing is invariant under link homotopy. Another result of this section says that simple loops in a blanket behave a lot like polygons in the plane: the measures of their internal angles, at least, have the same sum as they would for a polygon of the same number of vertices.

The two sides of a link

Because links are paths and not their images, every link is oriented. Hence when a simple link cuts a blanket into two scraps, one of these lies to the left of the link, and one lies to the right. This may seem obvious, but it requires some justification. Since we know what left and right mean in a sheet, we use the covering map to give an orientation to the blanket.

Definition 3c.1. Let α be a simple link in the blanket M, and let $p: M \to S$ be the covering map. Let τ be a linear path that intersects $Im\, \alpha$ at the point $\tau(1) = \alpha(x)$ alone. We say τ contacts α **from the left** or **from the right** according to whether the path $p \circ \tau$ contacts $p \circ \alpha$ from the left or the right in S.

What one must prove is that τ contacts α from the left if and only if $\tau(0)$ lies in a particular scrap of $M - Im\, \alpha$. We call this scrap *left*(α), the **left side** or **left scrap** of α, and we call the other scrap *right*(α), the **right side** or **right scrap** of α.

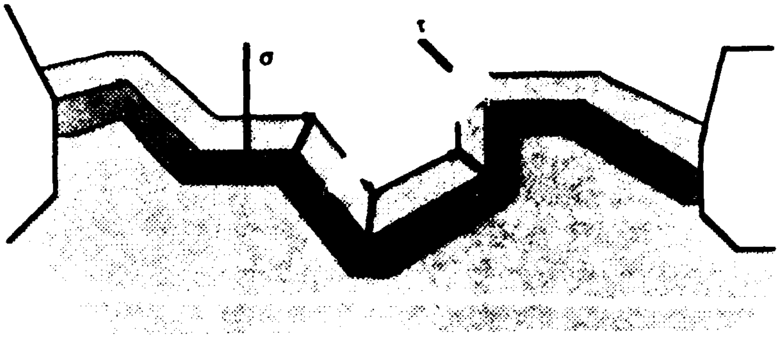


Figure 3c-1. *The two sides of a simple link in a blanket.* The link α separates the blanket M into two scraps, denoted *left*(α) and *right*(α). The shaded area represents a tubular neighborhood of α. The straight paths σ and τ both contact α from the left, and so σ can be moved to coincide with τ by a series of dilations, translations, and rotations. At each stage the origin of the path stays in the same scrap of $M - Im\,\alpha$. This construction shows that the left and right scraps of α are uniquely determined.

To prove that the left and right sides of α are well defined, one shows that if both σ and τ contact α from the left (or right), then σ can be moved along α until it coincides with τ. The construction, suggested by Figure 3c-1, uses a tubular neighborhood of α, and is rather messy. A similar idea underlies the following important proposition.

Proposition 3c.2. *Let α and β be simple links in a blanket, and suppose for some $e \in \{0,1\}$ that the points $\alpha(e)$ and $\beta(e)$ share a fringe. If β lies in left(α), then α lies in right(β), and we have the relations*

$$\mathit{left}(\beta) \subset \mathit{left}(\alpha) \qquad \text{and} \qquad \mathit{right}(\alpha) \subset \mathit{right}(\beta).$$

Proof. Let M denote the blanket. Choose x small enough that $\alpha_{e:x}$ and $\beta_{e:x}$ are straight, as shown in Figure 3c-2 below. Let κ be a simple path in $Bd\,M$ from $\alpha(e)$ to $\beta(e)$; it intersects $Im\,\alpha \cup Im\,\beta$ at its endpoints alone. Choose s and t so that $\kappa_{s:0}$ and $\kappa_{t:1}$ are straight. Because $\kappa(1)$ lies in *left*(α), so does $\kappa(s)$, and hence the path $\kappa_{s:0}$ contacts α from the left. Now let $p\colon M \to S$ be the covering map, and let F be the fringe of S containing $p \circ \kappa$. Let α' denote the path $p \circ \alpha_{e:x}$ if $e = 0$, and $p \circ \alpha_{x:e}$ if $e = 1$. Similarly define β'. By Definition 3c.1, the path $p \circ \kappa_{s:0}$ contacts α' from the left. Hence $p \circ \kappa$ traverses S in a counterclockwise direction if $e = 0$, or in a clockwise direction if $e = 1$. In either case, $p \circ \kappa_{t:1}$ contacts β' from the right. Therefore $\kappa_{t:1}$ contacts β from the right, which means that $\kappa(t)$ lies in *right*(β). Therefore $\kappa(0)$, and in fact all of $Im\,\alpha$, falls in *right*(β).

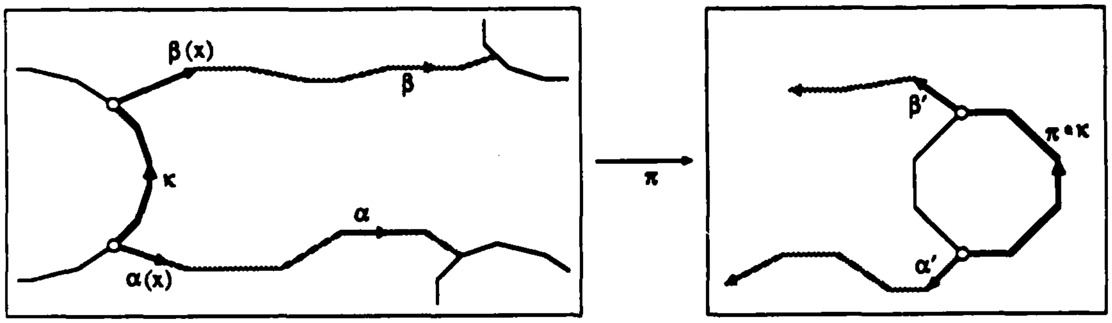


Figure 3c-2. *The construction that proves Proposition 3c.2.* **This figure shows the situation when $e = 0$. Because the reverse of the initial segment of κ contacts α from the left, the final segment of κ contacts β from the right.**

The desired inclusions now follow. Because $Im\,\alpha$ lies in $right(\beta)$, it does not intersect $left(\beta)$, and therefore the connected set $left(\beta)$ lies entirely in one scrap of $M - Im\,\alpha$. And since $left(\beta)$ contains points arbitrarily close to β, and β lies in $left(\alpha)$, the intersection $left(\beta) \cap left(\alpha)$ is nonempty. Thus we obtain $left(\beta) \subset left(\alpha)$. And this inclusion implies the other, because

$$right(\beta) = M - Im\,\beta - left(\beta) \supseteq M - left(\alpha) \supset right(\alpha). \quad \square$$

By symmetry, the claim remains true if we exchange *left* and *right* throughout.

Separations by homotopic links

A simple link in a blanket partitions the fringes of the blanket into three categories: those in its left side, those in its right side, and those it intersects (its terminals). How does this classification change when a link homotopy is applied to the link? The answer is that it remains unchanged. This fact follows fairly easily from Proposition 3b.8 if the links are disjoint, for then their images form a web of two threads. To deal with the possibility that the links intersect, I introduce one more method for constructing links. I call it the "detour lemma", because it constructs a simple link that detours around the right-hand sides of two given links.

Lemma 3c.3. (Detour Lemma) *Let α and β be link-homotopic simple links in a blanket M. There is a simple link γ in $Im\,\alpha \cup Im\,\beta$, link-homotopic to α and β, such that $right(\gamma)$ includes $right(\alpha)$ and $right(\beta)$.*

Proof. We construct γ by successive approximations. Begin with $\gamma = \alpha$, and let L and R be the left-hand and right-hand scraps of γ, respectively. Already γ satisfies all the conditions except $right(\gamma) \supseteq right(\beta)$. Because γ and β are piecewise linear, the path β protrudes into L only n times for some finite n. We proceed by induction on n, preserving all the conditions on γ except $right(\gamma) \supseteq right(\beta)$. In the basis case $n = 0$, the links α and β do not intersect. Since $\alpha(0)$ and $\beta(0)$ share a fringe, Proposition 3c.2 implies either $\mathit{right}(\alpha) \subset \mathit{right}(\beta)$ (if $\mathit{Im}\,\beta \subset \mathit{left}(\alpha)$) or $\mathit{right}(\beta) \subset \mathit{right}(\alpha)$ (if $\mathit{Im}\,\beta \subset \mathit{right}(\alpha)$). Choose $\gamma = \beta$ or $\gamma = \alpha$ accordingly.

Now suppose that $n > 0$. Let (s, t) be one of that open intervals that compose $\beta^{-1}(L)$. Splice the path $\beta_{s:t}$ into γ to form a simple link γ'. Let L' and R' be the left-hand and right-hand scraps of γ'. Because γ' shares some line segments of γ, the scraps R' and R intersect. Hence $R' \supseteq R$, since R is connected and does not intersect $\mathit{Im}\,\gamma'$. We also have $L' \subseteq M - R' - \mathit{Im}\,\gamma$, whence $L' \subsetneq L$. The containment is proper because $\beta_{s:t}$ lies in L but not L'. Replacing γ by γ', we reduce n by at least 1. Furthermore, the conditions on γ are maintained: we have $\mathit{Im}\,\gamma' \subseteq \mathit{Im}\,\alpha \cup \mathit{Im}\,\beta$; the terminals of γ' are those of α and β; and the right side R' of γ' includes R, which includes $\mathit{right}(\alpha)$ by assumption. The existence of the desired path γ follows by induction. □

Now we can prove the main result of this section.

Proposition 3c.4. *Link-homotopic simple links in a blanket partition the fringes identically.*

Proof. Let M be a blanket, with covering map $p\colon M \to S$, and let α and β be link-homotopic simple links in M. We first show that α and β may be assumed not to intersect, by finding a simple link δ that is link-homotopic to both α and β, but intersects neither of them. Apply Lemma 3c.3 to α and β, obtaining a simple link γ. The left-hand scrap of γ contains no points of $\mathit{Im}\,\alpha$ or $\mathit{Im}\,\beta$, else it would contain points in $\mathit{right}(\alpha)$ or $\mathit{right}(\beta)$, contradicting Lemma 3c.3. Let $\delta \in [\gamma]_L$ be a simple link in $\mathit{left}(\gamma)$. Then δ intersects neither α nor β, and since $\gamma \simeq_L \alpha$, we have $\delta \simeq_L \alpha$ as well.

We may therefore assume that the link-homotopic links α and β are disjoint. By Corollary 3a.5, α and β have the same terminals. Hence the set $\mathit{Im}\,\alpha \cup \mathit{Im}\,\beta$ is a web of 2 threads, because there is a loop of 2 links formed by α, a path in $\mathit{Bd}\,M$ from $\alpha(1)$ to $\beta(1)$, the reverse of β, and a path in $\mathit{Bd}\,M$ from $\beta(0)$ to $\alpha(0)$. By Lemma 3b.7, the set $M - (\mathit{Im}\,\alpha \cup \mathit{Im}\,\beta)$ has three components. One of these is a component of $M - \mathit{Im}\,\alpha$, one is a component of $M - \mathit{Im}\,\beta$, and the third (the inside of the web) contains no fringes, by Lemma 3b.8. Therefore α and β separate the fringes into the same three categories. Moreover, the fringes in the left scrap of α are also in the left scrap of β, because of Lemma 3c.2. □

The inside of a simple loop

According to Proposition 3b.8, each loop of links has an inside that contains no fringes. The same goes for simple loops in general, although we cannot say as much about the remaining components. One can prove this fact by analyzing an arbitrary simple loop in terms of loops of links.

Proposition 3c.5. *The image of a simple loop separates a blanket into two or more components, exactly one of which intersects no fringes.* □

The distinguished component is, of course, the **inside** of the loop. Since we are removing the entire image of the loop, and not just the threads it contains, the inside component actually avoids all fringes. We cannot claim that the components are scraps: if the loop touches no fringes, then its outside component is not simply connected.

Internal angles

Any bent path in a blanket makes an angle, and this angle can be measured by projecting it to the sheet. There is some ambiguity in this measurement, however: is the measure of the angle θ or $2\pi - \theta$? If the bent path is part of a simple loop, then the ambiguity can be resolved by considering the interior of the angle to be the side facing the inside of the loop. The resulting angle is called **internal angle** of the loop at that vertex. If the projection of the loop is a polygon with n vertices, we know by Euclidean geometry that the measures of the internal angles sum to $(n-2)\pi$. The same is true for any simple loop.

Lemma 3c.6. *If λ is a simple loop in a blanket, and λ has n vertices, then the measures of the internal angles of λ sum to $(n-2)\pi$.*

Proof. The proof is an induction that works by triangulating the loop. Let $\angle\lambda$ denote the sum of the measures of the internal angles of λ. The basis case is $n = 3$, when the projection of λ is a triangle. For the induction step, let $\lambda(a)$ and $\lambda(b)$ be the vertices adjacent to $\lambda(0)$, where $0 < a < b < 1$, and denote by m the measure of the internal angle formed by these three points. We can create and delete vertices of λ with measure π at will, for these operations change $\angle\lambda$ and $(n-2)\pi$ by the same amount. Hence we can assume $m \neq \pi$.

We find a linear path τ whose middle lies inside λ, and which divides λ into two loops with fewer than n vertices. See Figure 3c-3. If $m > \pi$, extend the linear path $\lambda_{a:0}$ into *inside*(λ) until it reaches a point $\lambda(t)$; let τ be the resulting linear path $\lambda(0) \triangleright \lambda(t)$. If $m < \pi$, let $T \subset I$ be set of values t for which there is a linear path from $\lambda(ta)$ to $\lambda(1 - t + tb)$ whose middle lies inside λ. If $1 \in T$, then let τ be the linear path $\lambda(a) \triangleright \lambda(b)$. Otherwise for $t = \sup T$ the middle of $\lambda(ta) \triangleright \lambda(1 - t + tb)$ intersects a vertex $\lambda(s)$ of λ; let τ be the linear path from $\lambda(0)$ to $\lambda(s)$.

In each case τ divides λ into simple loops μ and ν with fewer vertices than λ. If necessary, we create a vertex of λ at $\tau(1)$, so that the both endpoints of τ are vertices of λ. Then if λ has n vertices, μ has $k+2$ and ν has $n-k$. You can check that $0 < k < n-2$ in each of the cases (a), (b), and (c). The insides of the loops μ and ν cannot intersect *outside*(λ), else they would contain an entire component of

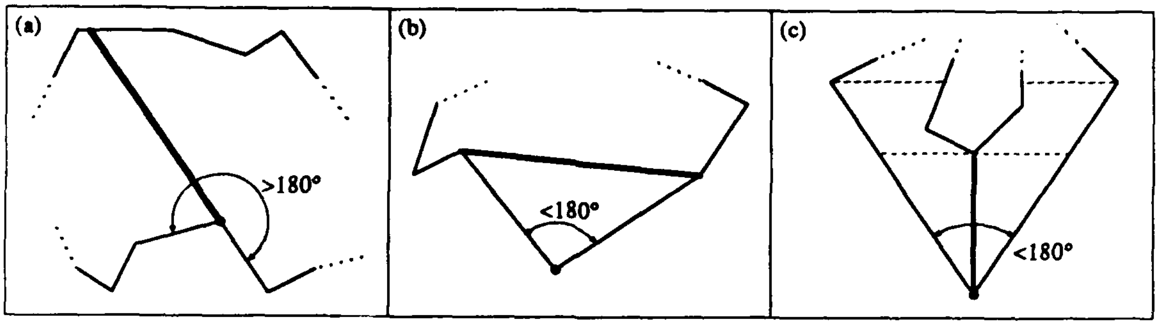


Figure 3c-3. *Triangulation of a simple loop.* Any simple loop in a blanket can be triangulated using these three operations: (a) extending an edge into the loop at an internal angle of measure $m > \pi$; (b) cutting off a triangle where the loop has an internal angle of measure $m < \pi$; and (c) if the linear path in part (b) does not exist or leaves the loop, dividing the internal angle with a linear path to the "nearest" other vertex.

$outside(\lambda)$ and hence intersect a fringe, in contradiction to Lemma 3c.5. It follows that every internal angle of μ and ν is part of an internal angle of λ, and thus $\angle\lambda = \angle\mu + \angle\nu$. The induction hypothesis now shows

$$\angle\lambda = (k+2-2)\pi + (n-k-2)\pi = (n-2)\pi,$$

and the proof is complete. □

Corollary 3c.7. *Every simple loop in a blanket has at least three internal angles of measure less than π.* □

3D. Elastic Chains in Sheets

Now we apply some of our results about blankets to paths in sheets. In Chapter 1 we saw the usefulness of rubber bands in sketches. The notion of a rubber band is even more natural in the sheet model, because the rubber band of a path need not leave the routing region. Recall that a **chain** for a path α is any path in $[\alpha]_P$. An **elastic** path is a canonical path α whose euclidean arc length is minimum among all paths in $[\alpha]_P$. The main result of this section is that every path has a unique elastic chain. It builds on two things: Lemma 3d.1 below, which says that every path can be made canonical without changing its path class, image, or arc length; and the results of the preceding section concerning loops in a blanket.

Parameterization of paths

The uniqueness result for elastic chains depends on the condition that an elastic chain be canonical. Without this restriction, all parameterizations of a minimum-length path would be elastic. The following lemma justifies our concentration on canonical paths; it shows that every path of finite arc length can be reparameterized to make it canonical.

Lemma 3d.1. (Reparameterization Lemma) *Let α be a path in R^n whose euclidean arc length $|\alpha|$ is finite. Then the map $f\colon s \mapsto |\alpha_{0:s}| \,/\, |\alpha|$ has a right inverse $g\colon I \to I$, and the function $\beta = \alpha \circ g$ is a canonical path with the same arc length as α. Furthermore,*

(1) *$\beta \simeq_P \alpha$ as paths in $Im\,\alpha$;*
(2) *β is piecewise linear if α is; and*
(3) *unless α is constant, β is not constant on any open interval of I.*

The function g is not necessarily continuous, but $\alpha \circ g$ is.

Proof. The function $f\colon I \to I$ defined by $f(s) = |\alpha_{0:s}| \,/\, |\alpha|$ is monotonic (non-decreasing) and continuous; it also satisfies $f(0) = 0$ and $f(1) = 1$. Hence f is surjective, so we can define a function $g\colon I \to I$ by $g(t) = \inf f^{-1}(t)$. Then g is monotonic because f is. Since $f^{-1}(t)$ is closed, we have $g(t) \in f^{-1}(t)$, which implies $f \circ g = id_I$. In other words, g is a right inverse of f. Put $\beta = \alpha \circ g$. We prove $\alpha = \beta \circ f$ by showing that for any $s \in I$, the path α maps s and $g(f(s))$ to the same point. Put $s' = g(f(s))$. Then $f(s) = (f \circ g \circ f)(s') = f(s')$, which means that $\alpha_{0:s}$ and $\alpha_{0:s'}$ have the same length. Hence $\|\alpha_{s:s'}\| = 0$, which means that α is constant on $[s', s]$.

To prove that β is continuous, let δ and t be given; we set $\epsilon = \delta / |\alpha|$ and show that $|t' - t| < \epsilon$ implies $|\beta(t') - \beta(t)| < \delta$. Put $s = g(t)$ and $s' = g(t')$. Then we have

$$|\alpha_{0:s}| = t \cdot |\alpha| \quad \text{and} \quad |\alpha_{0:s'}| = t' \cdot |\alpha| .$$

The difference between the left-hand sides of these equations is $|\alpha_{s:s'}|$, which is no less than the distance from $\alpha(s)$ to $\alpha(s')$. But these points are just $\beta(t)$ and $\beta(t')$, respectively. Thus

$$\begin{aligned} |\beta(t) - \beta(t')| &\le |\alpha_{s:s'}| \\ &= t' \cdot |\alpha| - t \cdot |\alpha| \\ &< \epsilon\, |\alpha| = \delta. \end{aligned}$$

Therefore β is a path.

Now we show that $|\beta_{0:t}| = t \cdot |\alpha|$ for arbitrary $t \in I$, thus proving that β is a canonical path with $|\beta| = |\alpha|$. We have $\beta(t) = \alpha(s)$ where $|\alpha_{0:s}| = t \cdot |\alpha|$, so it suffices to show that $\beta_{0:t}$ and $\alpha_{0:s}$ have the same arc length. By the definition of arc length, it is enough to show that $\beta_{0:t}$ and $\alpha_{0:s}$ have the same polygonal approximations. Let γ be a polygonal approximation to $\alpha_{0:s}$ with vertices $\alpha(s_0)$, $\alpha(s_1)$, $\ldots$, $\alpha(s_n)$; we have $s_0 = 0$ and $s_n = s$. The vertices of γ can also be written in the form $\beta(f(s_0))$, $\beta(f(s_1))$, $\ldots$, $\beta(f(s_n))$. Since f is a monotonic function satisfying

$f(0) = 0$ and $f(s) = t$, the path γ is also a polygonal approximation to β. Similarly, if γ is a polygonal approximation to β with vertices $\beta(t_0)$, $\beta(t_1)$, …, $\beta(t_n)$, then this sequence can be written $\alpha(g(s_0))$, $\alpha(g(s_1))$, …, $\alpha(g(s_n))$. Because g is a monotonic function satisfying $g(0) = 0$ and $g(t) = s$, the path γ is also a polygonal approximation to β.

Finally, we prove claims (1) through (3). The map f is a path in I from 0 to 1, and since I is simply connected, there is a path homotopy $F: I \times I \to I$ between f and the identity on I. Because $\beta = \alpha \circ f$, the map $\alpha \circ F$ is a path homotopy between β and α. Also $Im(\alpha \circ F) = Im\, \alpha$, so claim (1) is proved. For claim (2), suppose α is piecewise linear with vertices $\alpha(s_0)$, $\alpha(s_1)$, …, $\alpha(s_n)$. Then the function f is linear on each interval $[s_{i-1}, s_i]$, as is α, and so the map $\beta = \alpha \circ g$ is also linear on each interval $[f(s_{i-1}), f(s_i)]$. Since these intervals cover I, the path β is piecewise linear. For claim (3), suppose β is constant on some open interval (x, y). Then we have

$$0 = |\beta_{x:y}| = |\beta_{0:y}| - |\beta_{0:x}| = (y - x) \cdot |\alpha|\,,$$

so $|\alpha| = 0$, which implies that α is constant. □

Existence and uniqueness of elastic chains

Our results concerning elastic chains are established in five steps. The first step is a very intuitive one. It says that for a path to be minimal in length, all its subpaths must also be minimal.

Lemma 3d.2. *Every subpath of an elastic chain is elastic.*

Proof. Let β be an elastic chain, and let $\beta_{s:t}$ be a subpath of β. First of all, $\beta_{s:t}$ is canonical because for $x \in I$, we have

$$|(\beta_{s:t})_{0:x}| = \left|\beta_{s:s+x(t-s)}\right| = x \cdot |t - s| \cdot |\beta| = x \cdot |\beta_{s:t}|$$

since β is canonical. And if γ is path-homotopic to $\beta_{s:t}$, then the path β' defined by

$$\beta'_{0:s} = \beta_{0:s}, \quad \beta'_{s:t} = \gamma, \quad \beta'_{t:1} = \beta_{t:1}$$

is path-homotopic to β, and its euclidean arc length differs from that of β by $|\gamma| - |\beta_{s:t}|$. Because β is elastic, we have $|\beta'| \geq |\beta|$, and hence $|\gamma| \geq |\beta_{s:t}|$. Hence $\beta_{s:t}$ has minimum length among all paths in its path-homotopy class. □

The second step provides an important special class of elastic paths.

Lemma 3d.3. *A linear path is the unique elastic path in its path class.*

Proof. Let α and σ be path-homotopic elastic paths, and suppose σ is linear. Put $l = |\alpha| = |\sigma|$. Then $l = |\sigma| = |\alpha(0) - \alpha(1)|$ because σ is linear and has the same endpoints as α. For $t \in I$, we have

$$|\alpha(t) - \alpha(0)| \leq |\alpha_{0:t}| = tl,$$
$$|\alpha(t) - \alpha(1)| \leq |\alpha_{t:1}| = (1-t)l.$$

Thus $\alpha(t)$ lies within tl units of $\alpha(0)$ and $(1-t)l$ units of $\alpha(1)$. Only one point does so, namely $t\alpha(0) + (1-t)\alpha(1)$, which is $\sigma(t)$. Therefore $\alpha(t) = \sigma(t)$, and this identity holds for all t. □

The third step is the construction of elastic chains by means of Proposition 2c.8 and Lemma 3d.1.

Lemma 3d.4. *Every path in a sheet has an elastic chain.*

Proof. Let α be a path in the sheet S. First we show that $[\alpha]_P$ contains a minimum-length path. Let Π be the set of paths in $[\alpha]_P$, and let l denote $\inf_{\rho\in\Pi}|\rho|$. If some path $\rho \in \Pi$ satisfies $|\rho| = l$, then we are done. Otherwise by Proposition 2c.8 there is a uniformly convergent sequence $\langle\rho_i\rangle_{i=1}^{\infty}$ of links in Π whose limit ρ satisfies $|\rho| \leq l$. Because every link in Π has the same endpoints as α, we have $\rho(0) = \alpha(0)$ and $\rho(1) = \alpha(1)$.

We prove that ρ and α are path-homotopic. Let M be the blanket of S, and let $\widetilde{\rho}$ be any lifting of ρ to M. By Lemma 3a.7, there are liftings $\widetilde{\rho}_k$ of the paths ρ_k that converge uniformly to $\widetilde{\rho}$. Because the inverse image of $\rho(0)$ under the covering map is discrete, and similarly for $\rho(1)$, the paths $\widetilde{\rho}_k$ must have the same endpoints as $\widetilde{\rho}$ for sufficiently large k. Hence $\widetilde{\rho}_k \simeq_P \widetilde{\rho}$, which implies $\rho_k \simeq_P \rho$, for sufficiently large k. Therefore $\alpha \simeq_P \rho$. Now by Lemma 3d.1 there is a canonical path $\beta \in [\rho]_P$ whose arc length is that of ρ. This path β is an elastic chain for α. □

The fourth step brings elastic chains into the universe of piecewise linear objects, where we can apply our previous results to them. Let α be a PL path in a sheet S, and let x be a joint of α. The sheet S **restrains** α at x if for all sufficiently small open intervals (s,t) containing x, the path $\alpha(s) \triangleright \alpha(t)$ leaves S. If S restrains α at x, then $\alpha(x)$ is a vertex of a fringe of S, and α turns at x. We say α is **tight in S** if S restrains α at each of its joints.

Lemma 3d.5. *Elastic chains are piecewise linear and tight.*

Proof. Let α be a path in a sheet S, and let ρ be any elastic chain for α. The lemma is trivial if ρ is constant, so assume otherwise. We show that for every $x \in I$, either ρ is straight at x or ρ is bent at x. In either case there is an interval $[s,t]$ containing a neighborhood of x such that $\rho_{s:t}$ is bent. Since I is compact, finitely many such intervals cover I, and it follows that ρ is piecewise straight. The key fact we use is that every point y in the sheet S has a neighborhood that is starlike about y.

Let x be a point of $[0,1]$, and choose a neighborhood $U \subset S$ of $\rho(x)$ that is starlike about $\rho(x)$. Because ρ is continuous, all points $s \in I$ sufficiently close to x satisfy $\rho(s) \in U$, implying that the linear path $\sigma = \rho(x) \triangleright \rho(s)$ lies in U. Because U is starlike, it is contractible and hence simply connected (Lemma 2a.8). Therefore σ and $\rho_{x:s}$ are path-homotopic (Lemma 2a.5). By Lemma 3d.3, the path σ is the unique elastic path in its path class. Since $\rho_{x:s}$ is elastic, by Lemma 3d.2, it follows that $\rho_{x:s} = \sigma$. And since ρ is canonical, its subpath $\rho_{x:s}$ is not constant, and so $\rho_{x:s}$ is straight. We conclude that ρ is straight at x if $x \in \{0,1\}$, and a little further reasoning shows that ρ is bent at x if $x \in (0,1)$. Thus ρ is piecewise straight.

Now let x be a joint of ρ; we show that S restrains ρ at x. Let (s,t) be an interval containing x such that $\rho_{s:t}$ is bent. I show that for some interval (s',t') with $x \in (s',t') \subseteq (s,t)$ the path $\rho(s') \triangleright \rho(t')$ does not run in S. Let C denote the convex hull of the points $\rho(s)$, $\rho(t)$, and $\rho(x)$. Because C is convex, it is simply connected. Hence if $C \subseteq S$, then the path $\sigma = \rho(s) \triangleright \rho(t)$ would be path-homotopic to $\rho_{s:t}$ as paths in S. Since $\rho_{s:t}$ and σ are both elastic, they would have to be equal. But x is a joint of S, and so $\rho_{s:t}$ cannot equal the linear path σ. Therefore $C \not\subseteq S$, which implies that some linear path between $\rho_{s:x}$ and $\rho_{x:t}$ leaves S. Since the interval (s,t) was arbitrary, we conclude that S restrains ρ at x. Thus ρ is tight in S. □

The fifth and final step establishes the uniqueness property. It also shows something more, namely that for canonical paths, tightness implies elasticity.

Lemma 3d.6. *Let κ be a canonical, tight chain for a canonical path σ. Then $\|\kappa\| \leq \|\sigma\|$, with strict inequality if $\|\cdot\| = |\cdot|$ and $\kappa \neq \sigma$.*

Proof. Let $\tilde\kappa$ and $\tilde\sigma$ be path-homotopic lifts of κ and σ. By Lemma 3a.1, we can assume that $\tilde\sigma$ is simple, or else $\|\sigma\|$ could be reduced without changing $[\sigma]_P$. The lifting $\tilde\kappa$ is also simple, because κ is tight. For if $\tilde\kappa$ were not simple, either two consecutive segments of $\tilde\kappa$ would overlap, or some subpath of $\tilde\kappa$ would form a simple loop, and $\tilde\kappa$ would have to turn toward the inside of this loop at least once (Corollary 3c.7). But since κ is tight, $\tilde\kappa$ only turns toward fringes, and there are no fringes inside a simple loop (Proposition 3c.5). Therefore both $\tilde\sigma$ and $\tilde\kappa$ are simple and canonical, and they have the same endpoints. It follows that if $\tilde\kappa$ and $\tilde\sigma$ have the same image, then the two paths are equal. In this case $\kappa = \sigma$ and we are done. So we assume $Im\,\tilde\kappa \neq Im\,\tilde\sigma$ and prove $\|\kappa\| \leq \|\sigma\|$ with strictness if $\|\cdot\| = |\cdot|$.

Let (a,s) be the first crossing at which $\tilde\sigma$ leaves $Im\,\tilde\kappa$, and let (b,t) be the next crossing at which they rejoin. Then the paths $\tilde\kappa_{a:b}$ and $\tilde\sigma_{t:s}$ intersect at their endpoints alone, and their concatenation is a simple loop λ. We find a linear path in the blanket from $\tilde\sigma(s)$ to a point $\tilde\sigma(x)$; it will share a segment with $\tilde\kappa$. Because the blanket is simply connected, this path will be path-homotopic to $\tilde\sigma_{s:x}$. If we replace $\tilde\sigma_{s:x}$ by $\tilde\sigma(s) \triangleright \tilde\sigma(x)$, its arc length in the norm $\|\cdot\|$ will not increase; if $\|\cdot\|$ is the euclidean norm, then its arc length will actually decrease. Furthermore, $\widetilde{\sigma}$ will share one more segment of $\widetilde{\kappa}$. By repeated modifications of this kind, the path $\widetilde{\sigma}$ will converge to $\widetilde{\kappa}$.

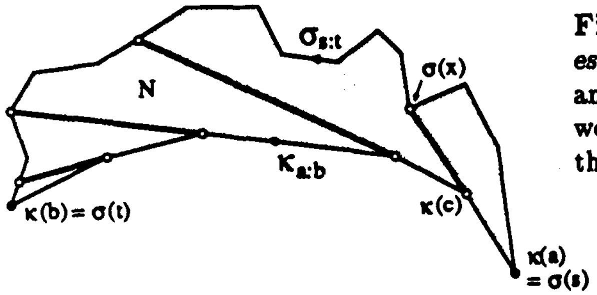


Figure 3d-1. *Why elastic chains are shortest in every norm.* Wherever the paths $\widetilde{\kappa}$ and $\widetilde{\sigma}$ form a loop, as here with $\widetilde{\kappa}_{a:b} = \widetilde{\sigma}_{s:t}$, we have $\|\kappa_{a:b}\| \leq \|\sigma_{s:t}\|$ by repeated use of the polygon inequality.

Let N denote the inside component of the simple loop $\lambda = \widetilde{\kappa}_{a:b} \star \widetilde{\sigma}_{t:s}$. Because κ is tight, its lift $\widetilde{\kappa}$ cannot turn toward N at any point in (a, b). If $\kappa_{a:b}$ is straight, then we are done; put $x = t$. Otherwise let c be the first point in (a, b) at which κ turns, and extend the path $\widetilde{\kappa}_{a:c}$ linearly into N. Eventually it must hit N again, either at $\widetilde{\kappa}(x)$ for $x \in (c, b]$, or at $\widetilde{\sigma}(x)$ for $x \in (s, t]$. In the latter case, we have the desired linear path $\widetilde{\sigma}(s) \triangleright \widetilde{\sigma}(x)$. The former case is ruled out, for the resulting simple loop $\widetilde{\kappa}_{c:x} \star (\widetilde{\kappa}(x) \triangleright \widetilde{\kappa}(c))$ could turn toward its inside only at the two points $\kappa(x)$ and $\kappa(c)$, whereas Corollary 3c.7 requires three such turning points. □

Two important results follow from Lemma 3d.6.

Corollary 3d.7. *The elastic chain of a path α is the unique canonical, tight chain in $[\alpha]_P$.* □

Corollary 3d.8. *The elastic chain κ for a path σ satisfies $\|\kappa\| \leq \|\sigma\|$ for any norm $\|\cdot\|$.* □

Chapter 4

Flow Across Cuts and Half-Cuts

The results of the next four chapters concern a model of single-layer wiring based on the relation of link homotopy in sheets. This model represents a layer of an integrated circuit or printed circuit board by a structure called a *design*. The term 'design' should be taken in the sense of 'pattern' or 'drawing', not in the sense of 'specification'. Like a sketch, a design embodies only the geometry and topology of a circuit layer, and none of its functionality. Table 4-1 records the correspondence between the elements of the design model and those of the sketch model. Logically, the design model is prior to the sketch model in that all my results about sketches are justified by relating them to analogous results about designs.

The purpose of this chapter is to lay the groundwork for the constructions and theorems that characterize routability and optimal routings of designs. (We will not reach those theorems until the middle of Chapter 6.) It begins by defining the design model and the concepts we use in analyzing it, and it proceeds to develop a detailed theory of the design model. This theory is not an outgrowth of any existing body of mathematics. It deals primarily with the properties and relations of cuts and wires that it invents. Nothing you have seen before will make its results obvious, although a familiarity with topology helps. It does, however, share with the sketch model a concern for the congestions and capacities of cuts, and the main results of this chapter can be understood in those terms.

As its title suggests, this chapter centers around the concept of *flow*. Flow is an abstraction that is similar to, but more versatile than, the concept of congestion we used up through Chapter 1. After defining the design model in Section 4A, we spend a section exploring the various equivalent definitions of flow and the relationship of flow to congestion. The flow across a cut is strongly related to the *necessary crossings* of the cut by wires, which we also define in Section 4B. We prove in Proposition 4b.3 that link-homotopic cuts have equal flow, and in Proposition 4b.3 that the flow across a simple cut equals its congestion. Later, in Section 4D, we define the concept of a *half-cut* for route of a wire, and extend the definition of flow to encompass half-cuts. We then prove an important formula (Proposition 4d.2) relating the flow across a cut to to the flows of the half-cuts it includes. Finally, Section 4F shows how to relate the flow and capacity of a cut to the flows and capacities in the links of a chain for that cut. We thereby obtain conditions under an unsafe simple cut or half-cut can be reduced to an unsafe *straight* cut or half-cut.

Comparing the two models

Sketches and designs differ in two major respects. First, we use the fringes of a sheet to represent the terminals and routing obstacles of a design, and hence these objects have positive size. Second, in a design we consider cuts that are not straight. Cuts and wires in the design model are links in a sheet, and they have the homotopy relation of links. Most terms, including *capacity*, *congestion*, *empty*, *entanglement*, *proper*, *routable*, *route*, *safe*, *self-avoiding*, *terminal*, and *width*, havc approximately the same meaning in both models.

Sketch model	Design model	Sketch Model	Design Model
feature	fringe	island	fringe
trace	wire	bridge	link
element	detail	bridge-homotopic	link-homotopic
realization	embedding	routing region	sheet
cut	straight cut	territory	extent

Table 4-1. *The correspondence between the sketch and design models.* Concepts that have the same name in both models are not shown.

The design model encompasses several ideas of what constitutes a proper design. To each there corresponds a routability theorem saying that a design is routable if and only if all of a certain class of straight cuts are safe. This class always excludes **trivial** cuts that are path-homotopic to paths in fringes. If one requires that the fringes of a proper design be self-avoiding, then the class includes all cuts with one terminal that are not trivial. If one allows the terminals of a wire in a proper design to be arbitrarily close, then the class excludes all cuts that are link-homotopic to wires. The most natural design model differs from the sketch model in these two ways. In order to support both models, we use the most permissive definition of a proper sketch, one in which fringes need not be self-avoiding and the terminals of a wire can be arbitrarily close.

4A. The Design Model

This section defines the design model and states the theorems that we set out to prove. These theorems, the *design routability theorem* and the *design routability* *theorem*, are the deepest results of the design model and the precursors of the corresponding theorems about sketches.

A design is essentially a set of disjoint simple links in a sheet, each one representing a wire. For technical reasons, however, we place some restrictions on these links and their terminals. A fringe F of a sheet S is called **inner** if $inside(F) \subset R^2 - S$, and otherwise F is **outer**. Every sheet has exactly one outer fringe and one or more inner fringes. A **wire** in a sheet S is a simple link in S with two convex inner fringes as terminals. A **design** on a sheet S, usually denoted Ω, is a finite set of wires in S whose images are disjoint and whose terminals are all distinct. The **details** of the design Ω are its wires and the fringes of S. An **article** of Ω is either a fringe of S that is not a terminal of Ω, or the union of the terminals and the image of some wire in Ω. Equivalently, an article of Ω is a component of the space $Bd\, S \cup \bigcup_{\omega \in \Omega} Im\, \omega$.

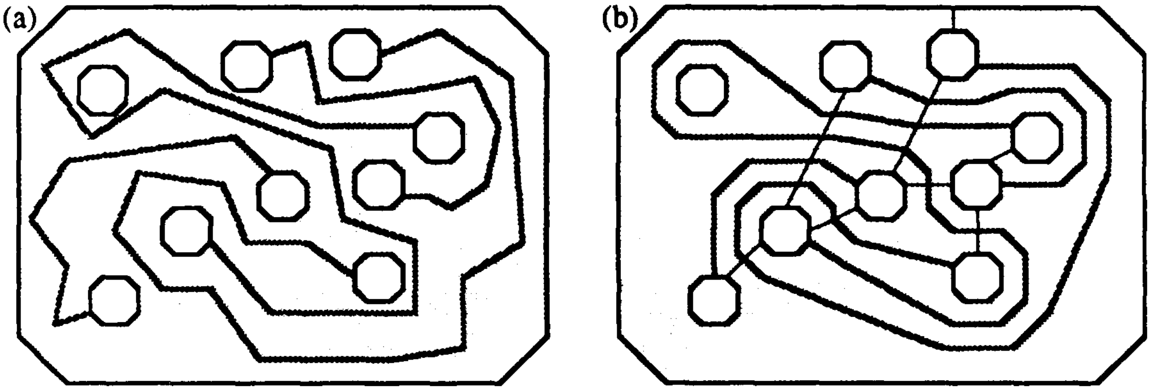


Figure 4a-1. *A design and one of its embeddings.* Panel (a) represents a 4-wire design on a sheet with 10 fringes. The dark polygons represent the fringes; the space inside the inner fringes and outside the outer fringe does not belong to the sheet. The inner fringes need not have the same shape, although they do in this example. Part (b) shows an embedding of the design at left: the sheets of the two designs are identical, and their wires are in bijective correspondence, with corresponding wires being link-homotopic.

Parallel to the concept of *realization* for sketches is the concept of *embedding* for designs. And as bridge homotopy governs the routing of traces, link homotopy governs the routing of wires. A link that is link-homotopic to a wire ω is called a **route** of ω. If this link is a wire, we call it an **embedding** of ω. If Ω and Υ are designs on the same sheet, we say Υ is an **embedding** of Ω if there exists a bijection $f: \Omega \rightarrow \Upsilon$ such that $\omega \simeq_L f(\omega)$ for every wire $\omega \in \Omega$. The embedding relation is an equivalence relation among the designs on a sheet.

The main problem concerning designs is that of finding a *proper* embedding for a design: an embedding that represents a legal circuit layer. As with a sketch, whether a design is proper depends upon the **widths** of its details. We assume that the design associates a positive width with each wire and fringe, with one important condition: no wire may be wider than either of its terminals. A route of a wire is always considered to have the same width as the original wire.

There are two ways a design can be improper. First, two of its articles may come too close. The **extent** of a detail F of width d is the set of points in R^2 lying within $d/2$ units of F. Distances here are measured by a piecewise linear **wiring norm**, denoted $\|\cdot\|$, that is a parameter of the entire model. The **extent** of an article is the union of the extents of its details. Different articles should have disjoint extents. Second, one of the wires of the design can have an undesirable shape. A subset X of R^2 is said to **divide** a sheet S if two fringes of S fall in different components of $R^2 - X$. An article of a design Ω on a sheet S is called **divisive** if its extent divides S. Every wire should be **self-avoiding**, meaning that its article should not be divisive.

To summarize: A design is **proper** if (1) its articles have disjoint extents, and (2) its wires are self-avoiding. A design is **routable** if it admits a proper embedding, and the wires in the proper embedding are called **feasible** embeddings of the wires in the original design.

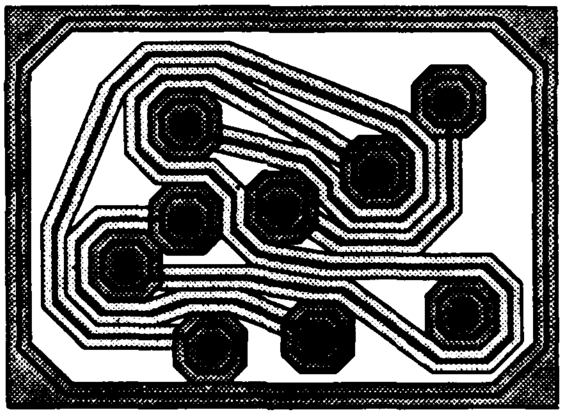


Figure 4a-.2. *The extents of a design's details.* This figure shows the thicknesses of the wires and fringes of the design in Figure 4a-1. As this drawing makes clear, the wires in that design were not routed arbitrarily. In fact, the embedding shown in Figure 4a-1 is optimal with respect to a certain octagonal norm, namely that in which each inner fringe is the set of points of distance 1 from its center of symmetry. By 'optimal' I mean that the embedding is proper and that no other proper embedding improves on the length of any wire.

Cuts and crossings

We analyze the routability of a design in terms of the congestions and capacities of cuts. The definition of cut in the design model is very general: a **cut** of a sheet S

is a link in S whose liftings to the sheet's blanket are simple. (Because the liftings of a link are related by covering transformations, either all the liftings of the link are simple, or none are.) Thus all simple links in the sheet, and all straight links in particular, are cuts. Let χ be a cut of S, and let Ω be a design on S. If the terminals of χ are X and Y, then the **capacity** of χ in Ω is

$$cap(\chi, \Omega) = \|\chi\| - width(X)/2 - width(Y)/2,$$

where $\|\chi\|$ is the arc length of χ in the wiring norm. We often abbreviate the notation $cap(\chi, \Omega)$ to $cap(\chi)$, for we shall never compare two designs that assign different widths to fringes.

Before defining the congestion of a cut, we need a precise notion of *crossing*. Since cuts can have self-intersections, we must count crossings according to multiplicity. If α and β are paths, a **crossing** of α by β is a pair $(s,t) \in I \times I$ such that $\alpha(s) = \beta(t)$. The pair $(s.t)$ is ordered; (t,s) would be a crossing of β by α. The number of crossings between α and β is denoted $cross(\alpha, \beta)$. Of course, the set of crossings can be infinite or even uncountable, but in the cases of interest it will be finite. The **entanglement** of a cut χ by a wire ω is defined to be the minimum number of crossings of χ by a route of ω. In symbols,

$$tangle(\chi, \omega) = \min\{\, cross(\chi, \omega') : \omega' \simeq_L \omega \,\}.$$

Because cuts are piecewise linear, entanglement is always finite. The **congestion** of χ in the design Ω, denoted $cong(\chi, \Omega)$, is the total entanglement of χ by wires in Ω, where each crossing is weighted according to the width of its wire. Formally, we have

$$cong(\chi, \Omega) = \sum_{\omega \in \Omega} width(\omega)\, tangle(\chi, \omega).$$

A simple cut is called **unsafe** if its congestion exceeds its capacity, and **safe** otherwise. Safety for nonsimple cuts is defined in Section 4F.

The intuitive meaning of congestion is this: If χ is a simple cut in a design Ω, then in any proper embedding of Ω, the portion of χ within the extents of wires will have total arc length at least $cong(\chi, \Omega)$. If this quantity is positive, and exceeds the capacity of χ, then no proper embedding of Ω can exist. Similarly, if the capacity of χ is negative, then the terminals of χ have overlapping extents. If these terminals lie in different articles, then Ω is again unroutable.

These and similar considerations motivate our definition of a *major* cut, one whose safety is necessary for the design to be routable. We say that a link χ is **degenerate** in Ω if χ is path-homotopic to a path in a single article of Ω. A cut χ is **empty** in Ω if $cong(\chi, \Omega) = 0$ and χ has only one terminal. Degenerate and empty cuts are called **minor**; others are **major**. The thin lines in Figure 4a-1 aie major straight cuts whose flow and capacity are equal. If any of these cuts were shorter, that design would be unroutable.

Central results concerning designs

I prove two major theorems in the design model: one concerns routability, and the other concerns routing. Chapter 8 uses these two theorems to prove the sketch routability theorem and the sketch routing theorem of Section 1A. The definitions in this section are arranged so as to permit a very simple characterization of routable designs.

Theorem 6c.1. (Design Routability Theorem) A design Ω on the sheet S is routable if and only if every major straight cut in S is safe in Ω.

If every major straight cut in S is safe in Ω, we say that Ω is **safe**. The design routability theorem has two parts: safe designs are routable (Theorem 5e.6), and unsafe designs are unroutable (Theorem 6a.5). The latter claim is the easier, and is proved in Section 6A.

The hard direction of the design routability theorem follows from a deeper result. It depends on the construction, presented in Section 5A, of an *ideal* embedding of every wire in a safe design. The ideal embedding of a wire is the shortest route *for that wire that leaves enough space for other wires to be routed. Formally, it* has minimum euclidean arc length among all routes for the wire whose *nontrivial, straight half-cuts are safe.*

Theorem 6c.2. (Design Routing Theorem) The ideal embeddings of the wires in a safe design form a proper design, and they have minimal euclidean arc length among all feasible embeddings of those wires.

In other words, when routing a safe design one can do no better than to use the ideal embedding of each wire. The proof of the design routing theorem occupies Chapter 5 and Section 6B.

4B. Flow: A Characterization of Congestion

Thanks to Chapters 2 and 3, *we already have many tools for examining designs.* We use them here to define formally the concept of a *necessary crossing*. As a consequence we are able to make sense of the congestion of nonsimple cuts. We characterize the congestion of a simple cut in terms of its necessary crossings by wires, and derive a statistic called the *flow* across a cut which agrees with congestion for simple cuts. The definition of flow turns out to be much more useful than the original definition of congestion, in part because it makes sense for cuts that are not simple, and in part because the topological machinery of Chapter 3 applies powerfully to the liftings and crossings that define flow. This power shows up immediately in the proof of Proposition 4b.3, which says that link-homotopic cuts have equal flow.

Necessary crossings in blankets

Intuitively, a necessary crossing between two links is one that cannot be removed by a link homotopy. Given two links in a blanket, one can tell whether they necessarily cross by examining their fringes.

Definition 4b.1. A simple link α in a blanket M **cuts** another link β in M if
(1) the endpoints of α and β lie on four distinct fringes of M, and
(2) the endpoints of β lie in different scraps of $M - Im\,\alpha$.

If α cuts β, then $Im\,\beta$ must intersect $Im\,\alpha$. For $Im\,\beta$ is a connected set; if it did not intersect $Im\,\alpha$ it would lie entirely in one component of $M - Im\,\alpha$. Furthermore, whether or not α cuts β depends only on the terminals of β, and not on any other properties of β. Hence if β is link-homotopic to another link β', then

$$\alpha \text{ cuts } \beta \iff \alpha \text{ cuts } \beta',$$

since (by Corollary 3a.5) β and β' have the same terminals. Thus if α cuts β, they make a crossing that cannot be removed by applying a link homotopy to β.

On the other hand, if α does not cut β, the crossing (if any) between α and β *can* be removed by applying a link homotopy to β. For if α does not cut β, then either (1) β shares a terminal with α, or else (2) the terminals of β lie on the same side of α. In either case, there is a link β' with the same terminals as β but whose endpoints lie in the same scrap of $M - Im\,\alpha$. By Proposition 3a.3 we can assume that β' is a link in that scrap, so that α and β' do not cross. Corollary 3a.5 implies that $\beta' \simeq_L \beta$. Thus the relation 'α cuts β' captures the intuitive notion that "β makes a necessary crossing with α".

The cutting relation has several other nice properties. If α' is simple and link-homotopic to α, then

$$\alpha \text{ cuts } \beta \iff \alpha' \text{ cuts } \beta,$$

because homotopic simple links separate the fringes identically (Proposition 3c.4). Moreover, if both α and β are simple, then the relation 'α cuts β' is symmetric. For if α does not cut β, then as shown above, some link $\beta' \in [\beta]_L$ lies in a single scrap of α. Clearly β' does not cut α, because their images are disjoint. Hence β does not cut α. We conclude that when α and β are simple,

$$\alpha \text{ does not cut } \beta \implies \beta \text{ does not cut } \alpha,$$

and the converse also holds by symmetry. Hence 'α cuts β' is a symmetric relation if α and β are simple.

Necessary crossings in sheets

The notion of necessary crossing for links in a blanket carries over to links in a sheet. To determine whether a crossing between two links in a sheet is necessary, we lift those links to the blanket in such a way that the lifts cross at the same point the original links cross, and check whether one lift cuts the other. The elegance and usefulness of this definition are two major motivations for using blankets to study wire routing.

Definition 4b.2. Let ω be a link in a sheet S, and let M be the blanket of S with covering map $p: M \rightarrow S$. Let χ be cut in S, and let $\tilde{\chi}$ be any lift of χ to M. Suppose that (s,t) is a crossing of χ by ω. Because $p(\tilde{\chi}(s)) = \omega(t)$, the link ω has a unique lift $\tilde{\omega}$ such that $\tilde{\chi}(s) = \tilde{\omega}(t)$. We say that $\tilde{\chi}$ and $\tilde{\omega}$ **reflect** the crossing (s,t). The crossing (s,t) of χ by ω is **necessary** if $\tilde{\chi}$ cuts $\tilde{\omega}$. Two crossings of χ by ω are **similar** if the corresponding lifts of ω are identical.

The initial choice of $\tilde{\chi}$ is irrelevant; it amounts to a choice of base point for the blanket, and as shown in Section 2B this choice does not affect the topology. If one chooses two different lifts of χ, say $\tilde{\chi}$ and $\tilde{\chi}'$, then one obtains different lifts $\tilde{\omega}$ and $\tilde{\omega}'$ of ω, and Proposition 2b.7 gives us a covering transformation $h: M \rightarrow M$ such that $h \circ \tilde{\chi} = \tilde{\chi}'$ and $h \circ \tilde{\omega} = \tilde{\omega}'$. Since the relation '$\tilde{\chi}$ cuts $\tilde{\omega}$' depends only on topological properties of M, $\tilde{\chi}$, and $\tilde{\omega}$, which are preserved by the homeomorphism h, the link $\tilde{\chi}$ cuts $\tilde{\omega}$ if and only if $\tilde{\chi}'$ cuts $\tilde{\omega}$. Hence necessity for crossings is well defined, and by similar reasoning, similarity is also. The technique of lifting links to reflect certain crossings among them will appear in future definitions, and we shall normally take for granted the fact that the choice of the first lifting—though not the choice of later liftings—is immaterial.

A definition equivalent to Definition 4b.2 would hold $\tilde{\omega}$ fixed and vary $\tilde{\chi}$ according to the crossing.

By counting necessary crossings we obtain a measure of the entanglement of two links. Two immediate consequences of Definition 4b.2 are that similarity of crossings is an equivalence relation, and that two similar crossings are either both necessary or both unnecessary. We define the quantity $wind(\chi,\omega)$, the **winding** of χ and ω, to be the number of similarity classes of necessary crossings between χ and ω. For any lift $\tilde{\chi}$ of χ, it is the number of lifts of ω that are cut by $\tilde{\chi}$. (Each such lift makes crossings with $\tilde{\chi}$, and these crossings form a similarity class of necessary crossings of χ by ω; conversely, every similarity class corresponds to a particular lift of ω that is cut by $\tilde{\chi}$.) Equivalently, since cutting is symmetric, $wind(\chi,\omega)$ is, for

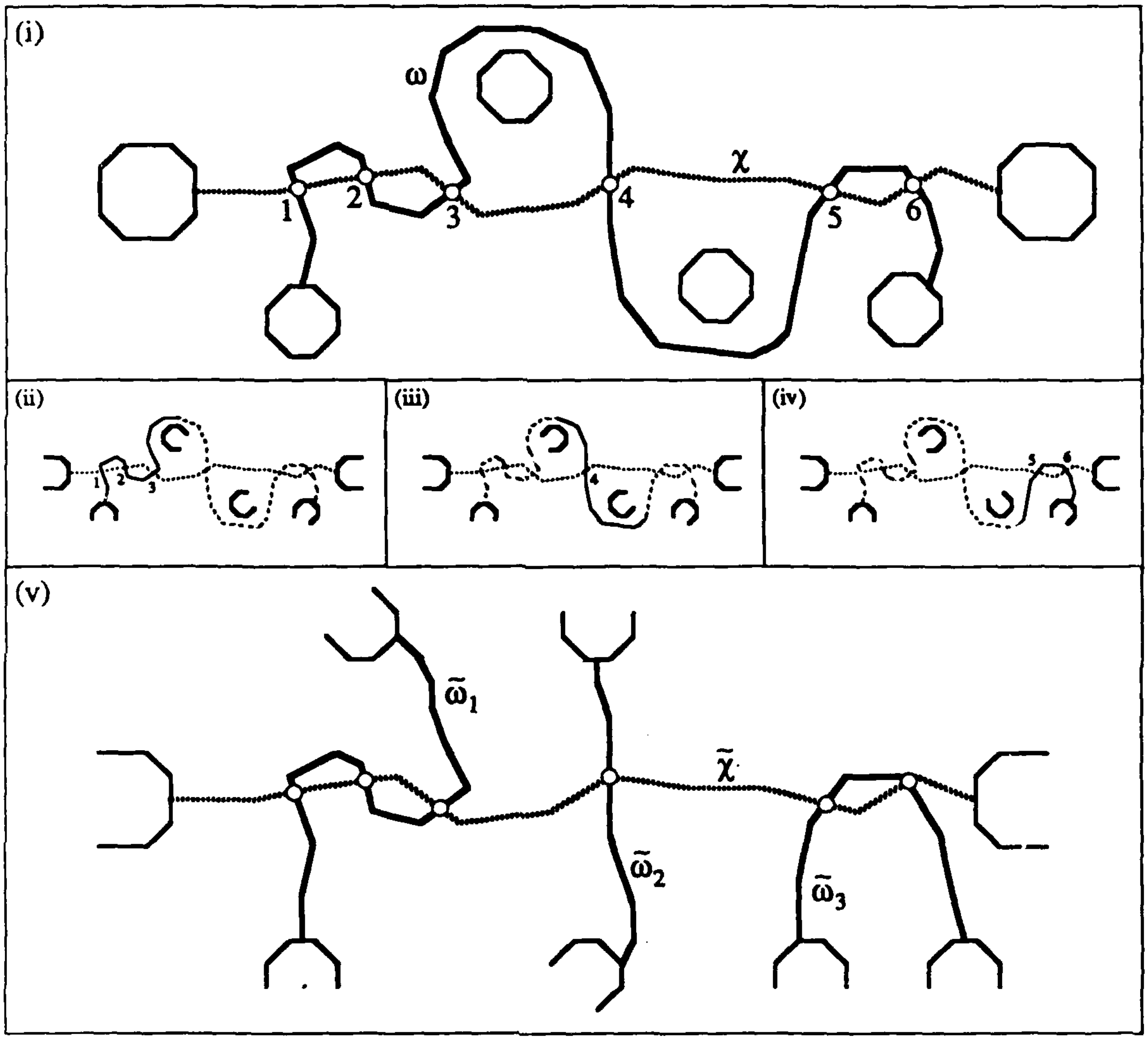


Figure 4b-1. *Necessity and similarity of crossings.* In part (i), the wire ω makes six crossings with the cut χ. The crossings fall into three similarity classes: crossings 1, 2, and 3 are similar and necessary; crossing 4 is necessary but not similar to the others; crossings 5 and 6 are similar and unnecessary. Parts (ii) through (iv) show the lifts of ω that correspond to these crossings, drawn in a fashion that emphasizes the covering map. Portions of these lifts are dotted to show that they run on a different level of the blanket from the lifting of χ. Part (v) summarizes the liftings of ω in a way that emphasizes their topology. Because ω is simple, these liftings do not intersect.

any lift $\tilde{\omega}$ of ω, the number of lifts of χ that cut, or are cut by, $\tilde{\omega}$. The winding of χ and ω in Figure 4b-1 is 2.

Summing the winding of χ over the wires in a design Ω, and weighting each number according to the width of the wire, we obtain a measure of congestion. I call it the **flow** across χ in the design Ω:

$$\mathit{flow}(\chi, \Omega) = \sum_{\omega \in \Omega} \mathit{width}(\omega)\, \mathit{wind}(\chi, \omega).$$

The flow statistic is invariant under link homotopy, both of wires and of cuts. Thus if Υ is an embedding of the design Ω, then $\mathit{flow}(\chi, \Upsilon) = \mathit{flow}(\chi, \Omega)$. Similarly, if α and β are link-homotopic cuts in the sheet of Ω, then $\mathit{flow}(\alpha, \Omega) = \mathit{flow}(\beta, \Omega)$. To emphasize this fact, I give it a formal proof.

Proposition 4b.3. *Link-homotopic cuts have equal flow.*

Proof. Let α and β be cuts of a sheet S, and let ω be a wire in S. Lift ω to the blanket of S, obtaining a link $\tilde{\omega}$. The flow of ω across α is the number of lifts $\tilde{\alpha}$ of α that cut $\tilde{\omega}$. Similarly, the flow of ω across β is the number of lifts $\tilde{\beta}$ of β that cut $\tilde{\omega}$. Assume now that α and β are link-homotopic. By Proposition 3a.6, there is a bijective correspondence becween the lifts of α and the lifts of β such that corresponding lifts are link-homotopic. Hence if a lift of α cuts $\tilde{\omega}$, so does the corresponding lift of β, and vice versa. Therefore $\mathit{wind}(\alpha, \omega) = \mathit{wind}(\beta, \omega)$. Since this holds for all wires ω, it follows that $\mathit{flow}(\alpha, \Omega) = \mathit{flow}(\beta, \Omega)$ for any design Ω on S. □

Proposition 4b.3 allows us to extend the concept of flow to all links. Since all cuts in a link-homotopy class have the same flow, and path-homotopic links are also link-homotopic, it suffices to define the flow of a link to be the flow of any path-homotopic cut. This works because every path class of links in a sheet contains a cut. For if α is a link with lifting $\tilde{\alpha}$, there is by Proposition 3a.3 a simple link $\tilde{\beta}$ between the endpoints of $\tilde{\alpha}$. By Lemma 2a.5, $\tilde{\beta}$ is path-homotopic to $\tilde{\alpha}$, and hence the projection β of $\tilde{\beta}$ to the sheet is path-homotopic to α. The link β is a cut because $\tilde{\beta}$ is simple.

Flow and congestion

We now address the question of how congestion compares to flow. The answer is that flow is never greater than congestion, and for simple cuts they are equal. The following two lemmas clarify the relationship between flow and congestion.

Lemma 4b.4. *Let χ be a cut of a sheet S. Every wire ω in S satisfies $\mathit{tangle}(\chi, \omega) \geq \mathit{wind}(\chi, \omega)$.*

Proof. Let M be a blanket of S with covering map $p: M \to S$. Denote by n the winding of ω and χ. Let $\tilde{\omega}$ be any lift of ω to M. Every necessary crossing of χ by ω represents a lift of χ that cuts $\tilde{\omega}$; dissimilar crossings correspond to different lifts of χ. Let $\tilde{\chi}_1, \ldots, \tilde{\chi}_n$ be the lifts of χ that cut $\tilde{\omega}$. Let v be any route of ω; we show that $cross(\chi, v) \geq n$, thus proving that $tangle(\chi,\omega) \geq n$. Using Proposition 3a.6, lift v to a link $\tilde{v} \in [\tilde{\omega}]_L$. Then for $1 \leq i \leq n$, the link $\tilde{\chi}_i$ cuts $\tilde{v}$, so we have $\tilde{\chi}_i(s_i) = \tilde{v}(t_i)$ for some $s_i, t_i \in I$. Projecting to the sheet, we see that each pair (s_i, t_i) is a crossing of χ by v. All these crossings are distinct. If $t_i = t_j$ for some i and j, then $\tilde{\chi}_i(s_i) = \tilde{\chi}_j(s_j)$, so by uniqueness of liftings, we cannot also have $s_i = s_j$ unless $i = j$. We conclude that $cross(\chi, v) \geq n$ as claimed. □

The other direction is somewhat harder, and it fails for cuts that are not simple, as shown in Figure 4b-2.

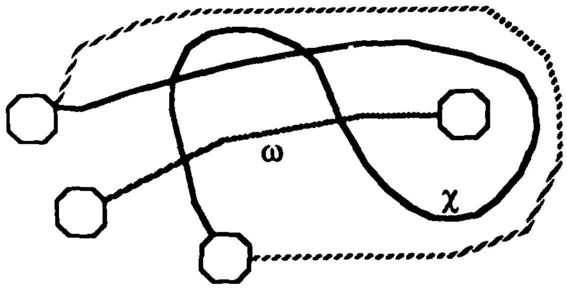


Figure 4b-2. *A cut whose flow and congestion differ.* The flow of ω across χ is zero, because χ is homotopic to a cut (striped path) that does not intersect ω. The entanglement of χ with ω is nonzero, however; every link that is homotopic to ω crosses χ at least twice.

Lemma 4b.5. *Let Γ be a set of disjoint simple cuts in a sheet S, and let ω be a link in S. There is a link $v \in [\omega]_L$ such that $cross(\gamma, v) = wind(\gamma, v)$ for all $\gamma \in \Gamma$.*

Proof. Let M be a blanket of S with covering map $p: M \to S$, and let $\tilde{\omega}$ be any lift of ω to M. Now let $\tilde{\gamma}_1, \ldots, \tilde{\gamma}_n$ be the links in M that lift elements of Γ and cut $\tilde{\omega}$. (Here n is $\sum_{\gamma\in\Gamma} wind(\gamma,\omega)$.) Two lifts $\tilde{\gamma}_i$ and $\tilde{\gamma}_j$ cannot intersect unless $i = j$. For if $\tilde{\gamma}_i(s) = \tilde{\gamma}_j(t)$, then $\tilde{\gamma}_i$ and $\tilde{\gamma}_j$ lift the same link γ, since the elements of Γ are disjoint. Thus $\gamma(s) = \gamma(t)$, whence $s = t$ because γ is simple, and thence $\tilde{\gamma}_i = \tilde{\gamma}_j$ by uniqueness of liftings. Figure 4b-3 illustrates the case where Γ contains but a single cut χ.

We construct a path $\tilde{v}$ that crosses each lifting $\tilde{\gamma}_i$ exactly once, and does not intersect any other lifting of any cut in Γ. In view of Lemma 4b.4, the conclusion will follow at once. Let A and B be the terminals of $\tilde{\omega}$. Denote by L_i and R_i the scraps of $M - Im\,\tilde{\gamma}_i$ that contain A and B, respectively. When $i \neq j$, the thread $Im\,\tilde{\gamma}_j$ must lie entirely in L_i or in R_i. By renumbering the lifts of γ, we may assume that $Im\,\tilde{\gamma}_j \in R_i$ whenever $j > i$. For each i such that $1 \leq i < n$, the set $L_{i+1} \cap R_i$ is a scrap—one component of $R_i - Im\,\gamma_{i+1}$.

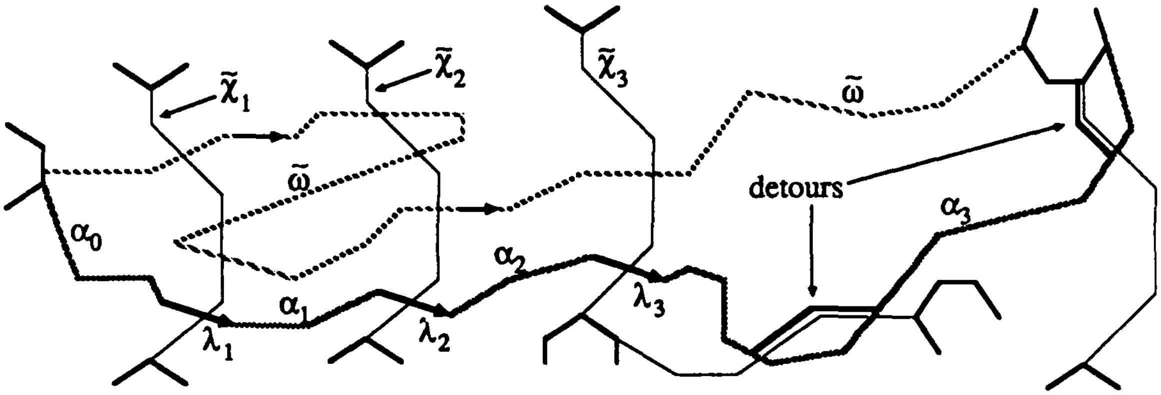


Figure 4b-3. *Why a simple cut's congestion does not exceed its flow.* **The n lifts of the simple cut χ that cut $\tilde{\omega}$ decompose the blanket into $n+1$ parts. One can construct a link in $[\tilde{\omega}]_L$ that crosses these lifts only once each: the concatenation of the paths α_i and λ_i is such a link. If this link crosses any other lifts of χ (thin lines), the crossings can be removed by inserting detours as shown.**

First we establish the points at which $\tilde{v}$ crosses the lifts $\tilde{\gamma}_i$. For $1 \leq i \leq n$, choose a straight path that crosses $p \circ \tilde{\gamma}_i$ exactly once and crosses no other cuts in Γ. Let λ_i be a lifting of either this path or its reverse, such that $\lambda_i(0) \in L_i$ and $\lambda_i(1) \in R_i$. Then λ_i makes exactly one crossing with a lifting of a cut in Γ; that lifting is $\tilde{\gamma}_i$. Now we connect up the segments λ_i with paths that intersect no liftings of cuts in Γ. By Proposition 3a.3, there is a simple half-link α_0 in L_1 from A to $\lambda_1(0)$, and a simple reverse half-link α_n in R_n from $\lambda_n(1)$ to B. For $i = 1, 2, \ldots, n$, use Proposition 3a.3 to find a simple path α_i in $L_{i+1} \cap R_i$ from $\lambda_i(1)$ to $\lambda_{i+1}(0)$. Define α to be the path

$$\alpha = \alpha_0 \star \lambda_1 \star \alpha_1 \star \cdots \star \lambda_n \star \alpha_n.$$

Then α crosses each of the lifts $\tilde{\gamma}_i$ exactly once. It may intersect some other lift of a cut in Γ, however.

Now we modify the subpaths α so that it intersects no liftings of cuts in Γ except $\tilde{\gamma}_1, \ldots, \tilde{\gamma}_n$. Using the fact that $Im\,\alpha$ is compact, one can check that it intersects only finitely many lifts of cuts in Γ. By induction, therefore, it suffices to show that a single unwanted crossing of α can be removed. Suppose that α crosses some lift $\tilde{\chi} \notin \{\tilde{\gamma}_1, \ldots, \tilde{\gamma}_n\}$ of a cut $\chi \in \Gamma$. Since $\tilde{\chi}$ is a simple link, it splits M into two scraps. And because $\tilde{\chi}$ does not cut $\tilde{\omega}$, at least one of these scraps contains portions of both A and B. Let N be such a scrap. Replace the portions of α that leave N by paths that skirt $\tilde{\chi}$ closely enough not to intersect any lifting of a cut in Γ. (Such a skirting path may be constructed using a tubular neighborhood of $Im\,\chi$ that intersects no other cut in Γ.) The resulting path is a still piecewise linear link from A to B, and it makes fewer crossings with lifts of cuts in Γ than it used to.

Eventually we obtain a piecewise linear link $\tilde{v}\colon A \rightsquigarrow B$ whose projection v makes at most n crossings with the cuts in Γ. Of course, the number of crossings it makes is actually n, by Lemma 4b.4. Corollary 3a.5 says that $\tilde{v}$ is link-homotopic to $\widetilde{\omega}$, and hence its projection v is link-homotopic to ω. □

If in Lemma 4b.5 we take Γ to be the set containing a single cut χ, we deduce that $\mathit{tangle}(\chi,\omega) \leq \mathit{wind}(\chi,\omega)$ whenever χ is simple. Combining this result with Lemma 4b.4, and summing over all the wires in a design, gives us the desired answer.

Proposition 4b.6. *If χ is a cut of a design Ω, then $\mathit{cong}(\chi,\Omega) \geq \mathit{flow}(\chi,\Omega)$, with equality if χ is simple.* □

Our interest in congestion comes from the design routability theorem (6c.1), which involves only straight cuts. Since congestion and flow agree for all simple cuts, we are free to discard the former in favor of the latter. And as Proposition 4b.3 suggests, flow is the more natural concept, and is far easier to work with. Henceforth we use the flow statistic exclusively, except in Chapter 8 when proving the sketch routability theorem.

4C. Relations Among Cuts and Wires

The main results of this chapter concern the flows across cuts. In order to relate the flows of different cuts in the same design, we first study relationships among simple links in a blanket. Of particular concern is the relation of one link cutting another, which forms the basis for the definition of flow. This section gives a condition under which one link must cut another, stated in Lemma 4c.1 below, and several conditions under which two links cannot cut one another, such as when they lift routes for wires in the same design.

We will use the following result many times.

Lemma 4c.1. *Let α and β be simple links in a blanket such that α cuts β, and let γ be a simple link from a terminal of α to a terminal of β. Every link that cuts γ also cuts either α or β.*

Proof. Without loss of generality we may replace α, β, and γ by link-homotopic simple links. We may also reverse α, β, and γ as desired. Choose simple links α and β that intersect in one point only, say $\alpha(s) = \beta(t)$, and let γ be the simple link $\alpha_{0:s} \star \beta_{t:1}$. By the symmetry between left and right we may assume that $\beta(1) \in \mathit{left}(\alpha)$, as in Figure 4c-1, and it follows that $\alpha(0) \in \mathit{left}(\beta)$.

I claim $\mathit{right}(\gamma) = \mathit{right}(\alpha) \cup \mathit{right}(\beta)$. The connected set $\mathit{right}(\alpha)$ does not intersect $\mathit{Im}\,\gamma$, but it borders on γ from the right at $\gamma(0)$. Hence $\mathit{right}(\gamma) \supseteq \mathit{right}(\alpha)$. Similarly $\mathit{right}(\beta)$ does not intersect $\mathit{Im}\,\gamma$, and it borders on γ from the right at $\gamma(1)$.

Hence $\mathit{right}(\gamma) \supseteq \mathit{right}(\beta)$. If a point x lies neither in $\mathit{right}(\alpha)$ nor $\mathit{right}(\beta)$, it must lie on $\mathit{Im}\,\alpha - \mathit{right}(\beta)$, or on $\mathit{Im}\,\beta - \mathit{right}(\alpha)$, or in $\mathit{left}(\alpha) \cap \mathit{left}(\beta)$. In the first two cases, x falls on $\mathit{Im}\,\gamma$. In the last case, draw a piecewise linear path from x to any point of $\mathit{Im}\,\alpha$ or $\mathit{Im}\,\beta$. The first point at which it intersects $\mathit{Im}\,\alpha \cup \mathit{Im}\,\beta$ must lie on $\mathit{Im}\,\gamma$, because $\mathit{Im}\,\alpha - \mathit{Im}\,\gamma \subset \mathit{right}(\beta)$ and $\mathit{Im}\,\beta - \mathit{Im}\,\gamma \subset \mathit{right}(\alpha)$, whereas x lies in $\mathit{left}(\alpha) \cap \mathit{left}(\beta)$. Of course, the path intersects α or β from the left. It follows that it intersects γ from the left, and hence x lies in $\mathit{left}(\gamma)$. We conclude that $\mathit{right}(\gamma) = \mathit{right}(\alpha) \cup \mathit{right}(\beta)$, and $\mathit{left}(\gamma) = \mathit{left}(\alpha) \cap \mathit{left}(\beta)$.

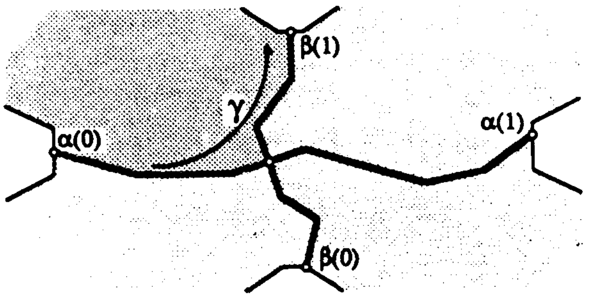


Figure 4c-1. *A link formed when two others cross.* The link γ, shown in grey, comprises parts of α and β. The left side of γ (dark shading) is the intersection of the left sides of α and β; the right side of γ (light shading) is the union of the right sides of α and β.

Now let η be a simple link that cuts γ. Then one terminal of η lies entirely in $\mathit{left}(\alpha) \cap \mathit{left}(\beta)$, and the other lies entirely in $\mathit{right}(\alpha) \cup \mathit{right}(\beta)$. Call the second terminal X. If X intersects $\mathit{right}(\alpha)$, then either it lies entirely in $\mathit{right}(\alpha)$, in which case η cuts α, or else it is a terminal of α. It cannot be the fringe containing $\alpha(0)$, because this is a terminal of γ. Hence X must be the fringe containing $\alpha(1)$, which lies wholly in $\mathit{right}(\beta)$. Then η cuts β. Similarly, if X intersects $\mathit{right}(\beta)$, then either $X \subset \mathit{right}(\beta)$ or else X is the fringe containing $\beta(0)$, which is a subset of $\mathit{right}(\alpha)$. In either case η cuts α or β. □

Liftings of wires and their routes

Many of the links we consider will be liftings of wires, or routes of wires, taken from the same design. Such links, if simple, are called *coherent*.

Definition 4c.2. Let Υ be a set of links obtained by replacing each wire in a design by a route of that wire. If the links in Υ have simple liftings, then any set of these liftings is called **coherent**. If α and β are simple liftings of links in Υ, then we say α **coheres with** β.

Coherent links do not cut one another. If they did, their projections to the sheet would have nonzero winding; and since winding is invariant under link homotopy, there would be two wires in a design with nonzero winding. But I claim that if ω and v are wires in a design, then $\mathit{wind}(\omega, v) = 0$. For if $\omega \neq v$, then ω and v do not intersect, and hence their lifts cannot intersect. Or if $\omega = v$, then since this link is simple, its lifts are all disjoint. In neither case can a lift of ω cut a lift of v, and thus $\mathit{wind}(\omega, v) = 0$.

Next we show that coherent links cannot have both terminals in common without being equal. This is a corollary of a result about link homotopy in blankets that is useful in its own right.

Lemma 4c.3. *If η is a link with two terminals, then no two distinct lifts of η are link-homotopic.*

Proof. Let α and β be link-homotopic lifts of η. We prove $\alpha = \beta$. By Corollary 3a.5, the points $\alpha(0)$ and $\beta(0)$ lie on the same fringe X, while $\alpha(1)$ and $\beta(1)$ lie on a different fringe Y. These fringes are different because η has two terminals. Let σ be a simple path in X from $\beta(0)$ to $\alpha(0)$, and let τ be a simple path in Y from $\alpha(1)$ to $\beta(1)$. The loop $\lambda = \alpha \star \tau \star \widehat{\beta} \star \sigma$ is inessential because the blanket M is simply connected.

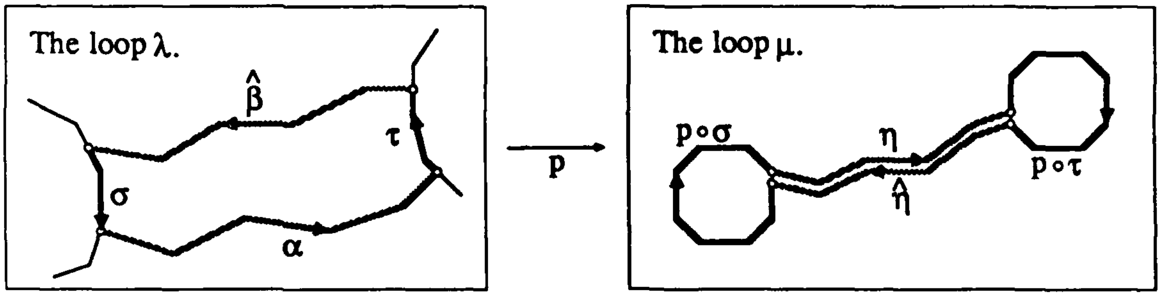


Figure 4c-2. *Link-homotopic lifts of a wire are identical.* For the fact that α and β are link-homotopic implies that the loop λ shown here is inessential, and consequently its projection μ is inessential. It would not be inessential if it wrapped around either terminal of ω, and so σ and τ are actually constant paths.

Now we project λ to the sheet S. Because λ is inessential in M, the resulting loop μ is inessential in S. Therefore μ is also inessential in the larger sheet $S' = S \cup \mathit{inside}(p(Y))$. Now

$$\mu = \eta \star (p \circ \tau) \star \widehat{\eta} \star (p \circ \sigma).$$

In S' the path $p \circ \tau$ is inessential, and hence μ is path-homotopic to $p \circ \sigma$ in S'. So $p \circ \sigma$ is inessential in S'. But $p \circ \sigma$ lies in $p(X)$, which by Lemma 3b.1 is a retract of S'. So $p \circ \sigma$ is inessential in $p(X)$, and hence in S. Therefore the endpoints of its lift σ are equal, which means $\alpha(0) = \beta(0)$. Hence $\alpha = \beta$ by uniqueness of liftings (Theorem 2b.2). □

Corollary 4c.4. *If α and β are unequal coherent links, then β has a terminal that is not a terminal of α.*

Proof. Let p denote the covering map. If $p \circ \alpha \neq p \circ \beta$, then these links are link-homotopic to distinct wires in a design. In this case all four terminals belonging to $p \circ \alpha$ and $p \circ \beta$ are different, and the same goes for α and β. On the other hand, if $p \circ \alpha = p \circ \beta$, then this link is homotopic to a wire, and hence has two terminals. Now Lemma 4c.3 shows that α and β are not link-homotopic. We cannot have $\alpha \simeq_L \widehat{\beta}$ either. Thus α and β have at least three terminals among them. □

The following lemma derives a further fact about coherent links. It will be needed in Chapter 5.

Lemma 4c.5. *If α and β are unequal coherent links, then the endpoints of β lie on the same side of α.*

Proof. Let p denote the covering map. If $p \circ \alpha \neq p \circ \beta$, then the terminals of $p \circ \alpha$ differ from those of $p \circ \beta$, and since coherent links do not cut one another, the endpoints of β must lie on the same side of α. We may therefore assume that α and β are liftings of the same path η, which is link-homotopic to a wire ω. If β does not share any terminals with α, we are done, because β does not cut α. So assume that $\alpha(0)$ and $\beta(0)$ lie on the fringe X. Then $\alpha(1)$ and $\beta(1)$ lie on different fringes, by Corollary 4c.4.

Because η is link-homotopic to a wire ω, Proposition 3a.6 provides lifts α' and β' of ω that are link-homotopic to α and β, respectively. Since α' and β' are distinct lifts of a simple path, they do not intersect, and hence the endpoints of β' lie on the same side, say the left, of α'. Let $F\colon \alpha \simeq_L \alpha'$ and $G\colon \beta \simeq_L \beta'$ be lifts of a link homotopy between η and ω, as in Lemma 3a.6. For every $t \in I$, the point $F(0,t)$ separates the fringe X into two components, a left component and a right component. We show that for every t, the point $G(0,t)$ lies to the left of $F(0,t)$. This is true at $t = 1$, because $G(0,1) = \beta'(1)$ and $F(0,1) = \alpha'(1)$. For no t are $F(0,t)$ and $G(0,t)$ equal, else $F = G$ by uniqueness of liftings and thus $\alpha = \beta$. Hence by continuity of F and G, the point $\beta(0) = G(0,0)$ lies to the left of $\alpha(0) = F(0,0)$. Also the fringe containing $\beta(1)$ lies in *left*(α) because it lies in *left*(α'). (Here we are using Lemma 3c.4.) Hence both endpoints of β lie left of α. □

A generalization of simplicity

Though the design routability theorem refers only to straight cuts, the results that lead to it use many cuts that are not straight, nor even simple. Often to make a proof go through one need not assume that a cut is simple, but rather that the cut *respects* the design in question. This relation is discussed in detail in Section 4E. For now we define a weaker relation, called *weak respect*, that is useful in obtaining upper bounds on flow.

Definition 4c.6. Let Ω be a design on a sheet S. A cut χ of S **weakly respects** the design Ω if whenever

(a) ω is a wire in Ω,
(b) $\tilde{\omega}$ and $\tilde{\omega}'$ are two lifts of ω that share a terminal, and
(c) $\tilde{\chi}$ is a lift of χ that cuts $\tilde{\omega}$,

the terminals of $\tilde{\omega}'$ lie within the same side of $\tilde{\chi}$.

For a cut to respect a design weakly means, in essence, that each fringe of the blanket contributes at most one necessary crossing to the flow across the cut. As one can check, this relation is invariant under link homotopy. In other words, if χ weakly respects a design Ω, then every cut χ' in $[\chi]_L$ weakly respects every embedding of Ω. Figure 4e-1 illustrates weak respect; Figure 4c-3 illustrates its absence.

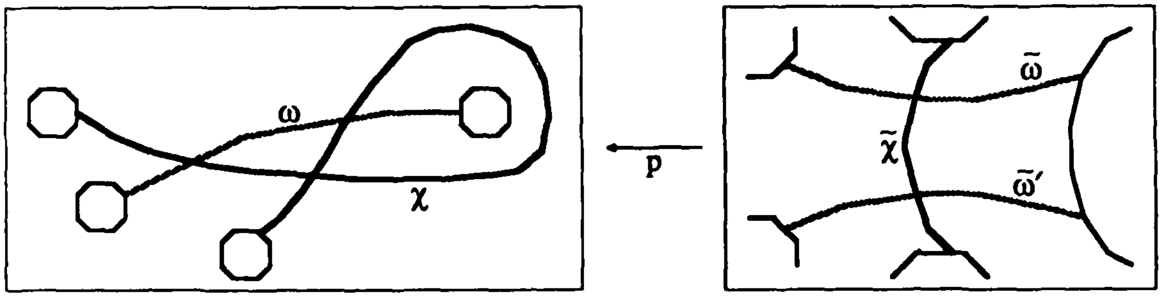


Figure 4c-3. *Lack of weak respect.* The cut χ (at left) does not weakly respect the wire ω (or rather, the one-element design $\{\omega\}$), because two lifts of ω that share a terminal cut the same lift of χ (at right).

If two lifts of a wire share a fringe, then some path from one to another wraps around the fringe one or more times. Figure 4c-3 shows a cut that does not weakly respect its design; it wraps around the terminal of a wire. This figure suggests that simple cuts have weak respect for all designs, a fact which we now prove.

Lemma 4c.7. *Simple cuts respect all designs weakly.*

Proof. Let M be a blanket on a sheet S, with covering map $p: M \rightarrow S$. Let χ be a simple cut of S, and let α be any lift of χ. Suppose that $\tilde{\omega}$ and $\tilde{\omega}'$ share a terminal and lift the same wire in S. We assume that α cuts $\tilde{\omega}$, and show that the terminals of $\tilde{\omega}'$ lie on the same side of α. For some $i, j \in \{0, 1\}$, the endpoints $\tilde{\omega}(i)$ and $\tilde{\omega}'(j)$ lie on the same fringe F. Because $p \circ \tilde{\omega} = p \circ \tilde{\omega}'$ is a wire, its terminals are distinct, and hence $i = j$; we may assume $i = j = 0$. Let $\alpha_1, \ldots, \alpha_n$ be the lifts of χ that cut $\tilde{\omega}$. (We have $n > 0$ because α cuts $\tilde{\omega}$.) Being distinct liftings of the simple path χ, the links $\{\alpha_i\}$ cannot intersect. Assume that α_1 is chosen so that $\alpha_2, \ldots, \alpha_n$ lie in the scrap of $M - Im\,\alpha_1$ that does not contain F. Let A denote the scrap of $M - Im\,\alpha_1$ that contains F.

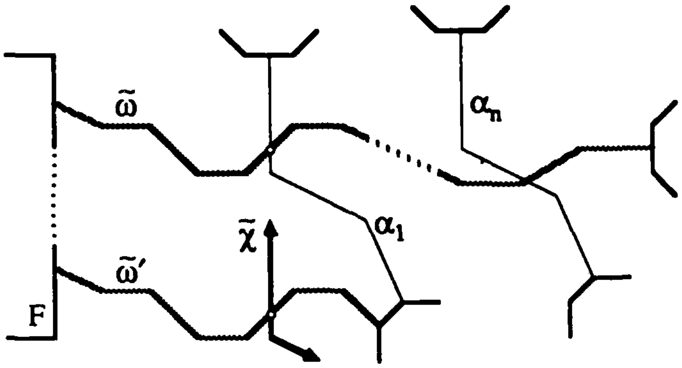


Figure 4c-4. *Why a simple cut respects all designs weakly.* The paths $\alpha_1, \ldots, \alpha_n$ are the lifts of a simple cut χ that cut the lift $\tilde{\omega}$ of the wire ω. Here $\tilde{\omega}'$ is another lift of ω that shares terminals with $\tilde{\omega}$ and α_1. The link $\tilde{\chi}$ is a lift of χ that is to $\tilde{\omega}'$ as α_1 is to ω. Where can it go? It must either cut $\tilde{\omega}$ or cross α_1, but it can do neither.

Because $\tilde{\omega}$ and $\tilde{\omega}'$ lift the same link ω, there is a covering transformation $h: M \to M$ such that $h \circ \tilde{\omega} = \tilde{\omega}'$. If h had a fixed point x, then h and id_M would be lifts of p agreeing at x, and Theorem 2b.2 would imply $h = id_M$. But h is not the identity transformation, because $\tilde{\omega} \neq \tilde{\omega}'$, so it has no fixed points, and hence $\alpha_1 \neq h \circ \alpha_1$. Because T is a homeomorphism, the lifts of χ that cut $\tilde{\omega}'$ are precisely $h \circ \alpha_1$, $\ldots$, $h \circ \alpha_n$. Also for this reason, and because $p \circ h = p$, the link $h \circ \alpha_i$ lies to the left of $h \circ \alpha_j$ whenever $i < j$. Note also that $h(F)$ is F.

We show that both terminals of $\tilde{\omega}'$ lie in A. Suppose not. Then either α_1 cuts $\tilde{\omega}'$, or else the two links share a terminal. In either case, $h \circ \alpha_1$ lies in A. (See Figure 4c-4.) For if α_1 cuts $\tilde{\omega}'$, then $\alpha_1 = h \circ \alpha_k$ for some $k > 1$, and $h \circ \alpha_1$ lies on the side of $h \circ \alpha_k$ that contains F, namely A. If instead α_1 shares a terminal with $\tilde{\omega}'$, that terminal cannot be F, and again $h \circ \alpha_1$ lies in A. Hence in either case, $h \circ \alpha_1 \neq \alpha_m$ for any m, so $h \circ \alpha_1$ does not cut $\tilde{\omega}$. Let $\eta \in [\tilde{\omega}]_L$ be simple and not intersect $h \circ \alpha_1$. The terminal F lies in A, so by assumption, the other terminal of $\tilde{\omega}'$ must have points outside A. By modifying the portion of η lying in $M - A$, we can obtain a simple link $\eta' \in [\tilde{\omega}']_L$ that does not intersect $h \circ \alpha_1$ either. Then $h \circ \alpha_1$ does not cut η', and hence does not cut $\tilde{\omega}'$. But we know that $h \circ \alpha_1$ does cut $\tilde{\omega}'$. This contradiction shows that the terminals of $\tilde{\omega}'$ must lie in A.

The rest is easy. Since $\alpha = \alpha_k$ for some k, neither terminal of α lies in A, while both terminals of $\tilde{\omega}'$ lie in A. Hence α cannot cut $\tilde{\omega}'$, and the two links cannot even share a terminal. Therefore the terminals of $\tilde{\omega}'$ lie on the same side of α. □

4D. Properties of Flow

The flow across a straight cut measures the total width of the wiring that must pass between two fringes. Another quantity that needs analysis is the amount of wiring that must pass between a fringe and a wire. To define it we introduce the concept of a *half-cut*, a half-link that begins on a fringe and ends on a wire.

This section defines half-cuts and explores their properties. Fortunately, we can study the attributes of half-cuts without introducing a lot of new concepts. Instead we define properties of half-cuts in terms of the properties of their *associated cuts*. In particular, the flow, degeneracy, triviality, and weak respect of a half-cut are defined in terms of the cut properties of the same names. (Much of the complexity of this theory, but also much of the interest, arises because the associated cuts of a half-cut are not, in general, simple, or even link-homotopic to anything simple.) In particular, we can relate the flows across half-cuts by analyzing flows across cuts. In this section we start to examine the methods for relating three or more cuts simultaneously. One important result is Proposition 4d.2. Whenever a wire (or a route of a wire) makes a necessary crossing with a cut, the half-cuts of the cut ending at that crossing have flows whose sum, when added to the width of the wire, equals or exceeds the flow across the cut.

Definition of a half-cut

Half-cuts arise as follows. Suppose ω routes a wire in a design Ω. Formally, a **half-cut for ω at t** is a half-link σ whose liftings are simple and which satisfies $\sigma(1) = \omega(t)$. Thus $\sigma(0)$ lies on a fringe and $\sigma(1) = \omega(t)$. For example, if (s,t) is a crossing of a cut χ by a link ω, then the half-links $\chi_{0:s}$ and $\chi_{1:s}$ are both half-cuts for ω at t. When the crossing (s,t) is clear from context we often omit mention of ω and t, and refer to $\chi_{0:s}$ and $\chi_{1:s}$ simply as half-cuts.

Attributes of half-cuts

Like cuts, half-cuts have flow and capacity in the context of a design. Let σ be a half-cut for ω at t, and suppose $\sigma(0)$ lies on the fringe F. The **capacity** of σ is defined to be

$$cap(\sigma,\omega) = \|\sigma\| - width(F)/2 - width(\omega)/2.$$

Since ω is a route of a wire in a particular design, the widths of ω and F are taken from this design. We sometimes abbreviate $cap(\sigma,\omega)$ to $cap(\sigma)$, even though σ alone does not specify which link ω is involved.

The flow across σ depends even more strongly on ω and on the crossing $(1,t)$ of σ by ω. If Ω is a design, we define $flow(\sigma,\Omega)$ to be $flow(\sigma \star \omega_{t:1},\Omega)$, which by

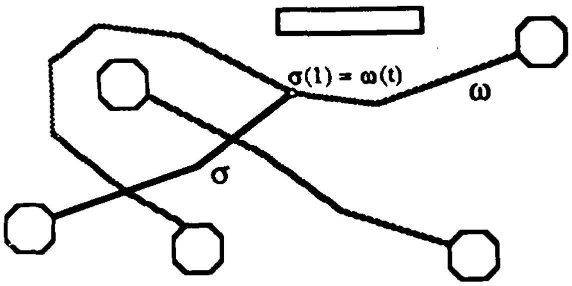


Figure 4d-1. *The flow across a half-cut.* The half-link σ is a half-cut for the wire ω at t. If the wires in this design have width 1, then the flow across σ, defined as the flow across the link $\sigma \star \omega_{t:1}$, is 2.

definition is $flow(\gamma, \Omega)$ where γ is any cut in $[\sigma \star \omega_{t:1}]_P$. Thus the flow across a half-cut is defined in terms of the flow across a cut.

This definition of flow makes intuitive sense, for if ω is a route of a wire in Ω, no wire in Ω can make a necessary crossing with ω. Hence the necessary crossings of $\sigma \star \omega_{t:1}$ must somehow reflect necessary crossings of σ. From a technical point of view the definition makes less sense, for two reasons. First, it can happen that no link in $[\sigma \star \omega_{t:1}]_L$ is simple, and thus we are forced to consider the flow across nonsimple cuts. Second, the choice of $\sigma \star \omega_{t:1}$ rather than $\sigma \star \omega_{t:0}$ is arbitrary, and yet significant: these two links can have different flows, even in a design consisting of ω alone.

Both technical difficulties can be overcome by extending the notion of weak respect (Definition 4c.6) to half-cuts. We do so by referring again to cuts. If σ is a half-cut for ω at t, then the cuts in the sets $[\sigma \star \omega_{t:0}]_L$ and $[\sigma \star \omega_{t:1}]_L$ are called **associated** to σ. The half-cut σ **weakly respects** a design Ω if every cut associated to σ weakly respects Ω. Since weak respect is invariant under link homotopy, this condition is not as restrictive as it sounds. Lemma 4d.3 below shows that if σ respects Ω weakly, then all associated cuts of σ have the same flow in Ω.

Associated cuts help us define other properties of half-cuts as well. For instance, we call a half-cut σ **degenerate** if it has a degenerate associated cut. Similarly, a half-cut is **trivial** if it has a trivial associated cut. Triviality can be cast in terms of liftings. Let σ be a half-cut for ω at t, and let $\widetilde{\sigma}$ and $\widetilde{\omega}$ be lifts of σ and ω that reflect the crossing $(1, t)$. In other words, $\widetilde{\sigma}(1) = \widetilde{\omega}(t)$. Then σ is trivial if and only if the terminal of $\widetilde{\sigma}$ is a terminal of $\widetilde{\omega}$.

Equivalence of half-cuts

If σ and τ are homotopic as half-links, meaning that $\sigma(1) = \tau(1)$ and $\sigma \star \widehat{\tau}$ is trivial, then σ is a half-cut for ω at t if and only if τ is. We also have $\sigma \star \omega_{t:1} \simeq_L \tau \star \omega_{t:1}$, and hence (by Proposition 4b.3) homotopic half-cuts have equal flow. There is, however, a much coarser equivalence relation on half-cuts that preserves flow.

Definition 4d.1. Let σ and τ be half-cuts for ω at s and v at t, respectively, where $\omega \simeq_L v$. Suppose $\widetilde{\omega}$ and $\widetilde{\sigma}$ are lifts of ω and σ such that $\widetilde{\sigma}(1) = \widetilde{\omega}(s)$. Also

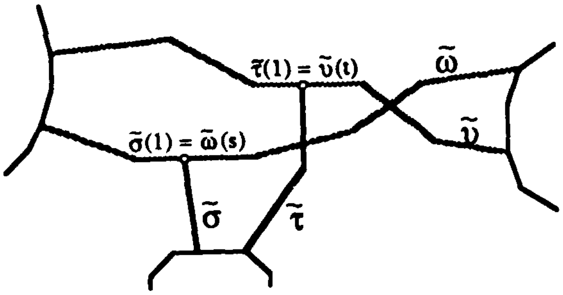

$\tilde{\tau}(1) = \tilde{v}(t)$
$\tilde{\omega}$
$\tilde{v}$
$\tilde{\sigma}(1) = \tilde{\omega}(s)$
$\tilde{\sigma}$
$\tilde{\tau}$

Figure 4d-2. *Equivalence of half-cuts.* A half-cut σ for ω at s is *akin to* a half-cut τ for v at t if there are lifts $\tilde{\sigma}$ and $\tilde{\omega}$ reflecting the crossing $(1, s)$, and lifts $\tilde{\tau}$ and $\tilde{v}$ reflecting $(1, t)$, that share fringes as shown here.

suppose $\tilde{v}$ and $\tilde{\tau}$ are lifts of v and τ such that $\tilde{\tau}(1) = \tilde{v}(t)$. We say that σ and τ are **akin** if the lifts may be chosen so that $\tilde{\omega} \simeq_L \tilde{v}$ and $\tilde{\sigma}$ and $\tilde{\tau}$ have the same terminal.

If σ and τ are akin, then Corollary 3a.5 implies that $\tilde{\sigma} \star \tilde{\omega}_{s:1}$ and $\tilde{\tau} \star \tilde{v}_{t:1}$ are link-homotopic, and hence their projections to the sheet are link-homotopic also. It follows from Proposition 4b.3 that σ and τ have the same flow in any design.

In fact, all the properties we define for half-cuts, except geometric quantities like capacity, are invariant under kinship—the relation of being akin. The reason is that half-cuts that are akin have link-homotopic associated cuts. Hence half-cuts that are akin are equally degenerate, and they have weak respect for the same designs.

Mid-cuts

Just as there are cuts from fringes to fringes and half-cuts from fringes to wires, there are mid-cuts from wires to wires. We shall occasionally have use for them. Suppose v and ω are routes of wires in a design Ω on the sheet S. For $s, t \in (0, 1)$, a **mid-cut** between v at s and ω at t is a mid-link τ in S whose liftings are simple and which satisfies $\tau(0) = v(s)$ and $\tau(1) = \omega(T)$. We define the properties of mid-cuts by analogy with half-cuts. The capacity of the mid-cut τ is

$$cap(\tau) = \|\tau\| - width(v)/2 - width(\omega)/2,$$

and its associated cuts are the cuts in the sets $[v_{i:s} \star \tau \star \omega_{t:j}]_L$, for $i, j \in \{0, 1\}$. A mid-cut respects a design weakly if all its associated cuts do, and it is degenerate if its associated cuts are. We define the flow across the mid-cut τ to be the flow across the link $v_{1:s} \star \tau \star \omega_{t:1}$. I leave it to the reader to adapt the definition of kinship (4d.1) to mid-cuts.

Together with cuts, half-cuts and mid-cuts are collectively known as **subcuts**. All liftings of subcuts are simple sublinks.

Combining two half-cuts

The main impact of the study of blankets is that it allows us to relate the flows of different cuts. These relationships are the theme of the rest of the chapter. Our first result relates the flow across a cut χ to the sum of the flows across half-cuts $\chi_{0:s}$ and $\chi_{1:s}$ for a link ω. It says that if the half-cuts lie on "opposite sides" of ω, then the flows across these half-cuts, and ω itself, all contribute to the flow across χ. This result was first claimed (in a different model) by Cole and Siegel, who used it in [6] without proof. Our knowledge of the topology of blankets allows us to give a rigorous proof of a more general claim.

Proposition 4d.2. *Let χ be a cut of a sheet S, and let Ω be a design on S. Suppose that (s,t) is a necessary crossing of χ by a route ρ of a wire ω in Ω. Then*

$$\mathit{flow}(\chi,\Omega) \geq \mathit{flow}(\chi_{0:s},\Omega) + \mathit{flow}(\chi_{1:s},\Omega) + \mathit{width}(\omega),$$

with equality if χ respects ω weakly.

Proof. Let M be a blanket for S, with covering map $p\colon M \to S$. Because the crossing (s,t) is necessary, there are lifts $\widetilde{\chi}$ of χ and $\widetilde{\rho}$ of ρ such that $\widetilde{\chi}(s) = \widetilde{\rho}(t)$, and $\widetilde{\chi}$ cuts $\widetilde{\rho}$. Let X and Y be the fringes of M containing $\widetilde{\chi}(0)$ and $\widetilde{\chi}(1)$, respectively, and let Z be the fringe of M containing $\widetilde{\rho}(1)$. Assume without loss of generality that $Z \subset \mathit{right}(\widetilde{\chi})$. Let $\widetilde{\alpha}$ be a simple link in $\mathit{right}(\widetilde{\chi})$ from X to Z. Then by Proposition 3c.2, we have $\mathit{Im}\,\chi \in \mathit{left}(\widetilde{\alpha})$. Let $\widetilde{\beta}$ be a simple link in $\mathit{right}(\widetilde{\chi}) \cap \mathit{left}(\widetilde{\alpha})$ from Y to Z. By Corollary 3a.5, we have the relations

$$\widetilde{\alpha} \simeq_L \widetilde{\chi}_{0:s} \star \widetilde{\rho}_{t:1} \quad \text{and} \quad \widetilde{\beta} \simeq_L \widetilde{\chi}_{1:s} \star \widetilde{\rho}_{t:1}.$$

Write $\alpha = p \circ \widetilde{\alpha}$ and $\beta = p \circ \widetilde{\beta}$. Projecting to the sheet, we have $\alpha \simeq_L \chi_{0:s} \star \rho_{t:1}$ and $\beta \simeq_L \chi_{1:s} \star \rho_{t:1}$. Hence by Proposition 4b.3 and the definition of the flow across a half-cut, we have $\mathit{flow}(\chi_{0:s},\Omega) = \mathit{flow}(\alpha,\Omega)$ and $\mathit{flow}(\chi_{1:s},\Omega) = \mathit{flow}(\beta,\Omega)$. Hence it suffices to prove

$$\mathit{flow}(\chi,\Omega) \geq \mathit{flow}(\alpha,\Omega) + \mathit{flow}(\beta,\Omega) + \mathit{width}(\omega), \tag{4-1}$$

with the reverse inequality

$$\mathit{flow}(\chi,\Omega) \leq \mathit{flow}(\alpha,\Omega) + \mathit{flow}(\beta,\Omega) + \mathit{width}(\omega) \tag{4-2}$$

holding also if χ respects Ω weakly.

Bounds on the flow across χ come from comparing the links that cross $\widetilde{\chi}$ to those that cross $\widetilde{\alpha}$ and $\widetilde{\beta}$. Every lift $\widetilde{v}$ of v that cuts $\widetilde{\chi}$ contributes exactly $\mathit{width}(v)$ to $\mathit{flow}(\chi,\Omega)$, and similar statements hold for $\mathit{flow}(\alpha,\Omega)$ and $\mathit{flow}(\beta,\Omega)$. So suppose that $\widetilde{v}$ lifts a wire $v \in \Omega$. We show that if $\widetilde{v}$ cuts $\widetilde{\alpha}$ or $\widetilde{\beta}$, then $\widetilde{v}$ cuts $\widetilde{\chi}$. By Proposition 3a.6, there is a lift $\widetilde{\omega}$ of ω in $[\widetilde{\rho}]_L$; its terminals are those of $\widetilde{\rho}$, and hence $\widetilde{\chi}$ cuts $\widetilde{\omega}$. Since $\widetilde{\alpha}$ runs from a terminal of $\widetilde{\chi}$ to one of $\widetilde{\omega}$, Lemma 4c.1 shows that if $\widetilde{v}$ cuts $\widetilde{\alpha}$, it must also cut either $\widetilde{\chi}$ or $\widetilde{\omega}$. Similarly, if $\widetilde{v}$ cuts $\widetilde{\beta}$, it must also cut either $\widetilde{\chi}$ or $\widetilde{\omega}$. But the links $\widetilde{v}$ and $\widetilde{\omega}$ cohere, and therefore they do not cut one another.

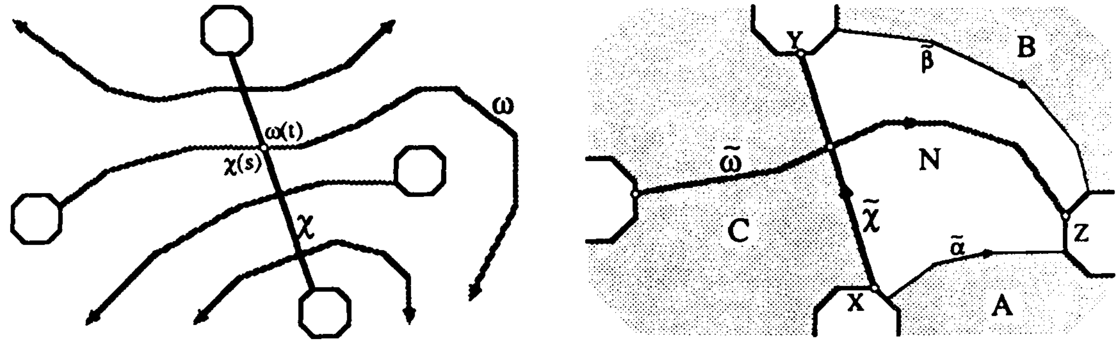


Figure 4d-3. *Combining half-cuts to form* a *cut.* At left, the straight half-cuts $\chi_{0:s}$ and $\chi_{1:s}$ for ω at t connect to form the straight cut χ. Because the crossing (s,t) of χ by ω is necessary, the lift $\tilde{\chi}$ cuts the lift $\tilde{\omega}$, at right. Every wire lifting that cuts $\tilde{\alpha}$ or $\tilde{\beta}$ also cuts $\tilde{\chi}$ because it cannot cut $\tilde{\omega}$. Conversely, every wire lifting that cuts $\tilde{\chi}$ also cuts $\tilde{\alpha}$ or $\tilde{\beta}$ unless it shares the terminal Z with $\tilde{\omega}$.

Hence every lift $\tilde{v}$ that contributes to the flow across α or β contributes the same amount to the flow across χ. This observation alone implies that $\mathit{flow}(\chi,\Omega)$ is no less than $\mathit{flow}(\alpha,\Omega) + \mathit{flow}(\beta,\Omega)$. The term $\mathit{width}(\omega)$ in inequality (4–1) is accounted for by the lift $\tilde{\omega}$ of ω. It cuts $\tilde{\chi}$, but does not cut $\tilde{\alpha}$ or $\tilde{\beta}$, since it shares the terminal Z with the latter links. So $\tilde{\omega}$ contributes an extra amount $\mathit{width}(\omega)$ to $\mathit{flow}(\chi,\Omega)$. Thus inequality (4–1) is established.

Now we suppose that χ weakly respects ω, and prove inequality (4–2). The threads $\mathit{Im}\,\tilde{\alpha}$, $\mathit{Im}\,\tilde{\beta}$, and $\mathit{Im}\,\tilde{\chi}$ form a web of 3 threads. By Lemma 3b.7 and Proposition 3b.8, they separate M into 4 scraps. Three of these, call them A, B, and C, border on the threads $\mathit{Im}\,\tilde{\alpha}$, $\mathit{Im}\,\tilde{\beta}$, and $\mathit{Im}\,\tilde{\chi}$, respectively; the fourth scrap, call it N, borders on all three threads, and it contains no fringes. Let $\tilde{v}$ lift any wire $v \in \Omega$. If $\tilde{v}$ cuts $\tilde{\chi}$, it must have one terminal in C, and its other terminal, if not Z, must lie in A, B, or N. It cannot lie in N, because N contains no fringes. And if it lies in A or B, then $\tilde{v}$ cuts $\tilde{\alpha}$ or $\tilde{\beta}$, respectively. Hence the only way that $\tilde{v}$ can contribute to the flow across χ without contributing to the flow across α or β is if $\tilde{v}$ has Z as a terminal and cuts $\tilde{\chi}$. And since different wires in a design have different terminals, this implies $v = \omega$. Because χ weakly respects ω, Definition 4c.6 says that no lift of ω other than $\tilde{\omega}$ can cut $\tilde{\chi}$ and have Z for a terminal. Inequality (4–2) follows. □

Respect and half-cuts

As previously mentioned, the definition of flow for half-cuts is somewhat arbitrary; the associated cuts of a half-cut can have quite different properties. In particular, some can respect a design without the others doing so, and they can have different flows, even considering only the half-cut's wire. These facts are illustrated by Figure 4d-4. But if a half-cut respects its wire weakly, this problem goes away.

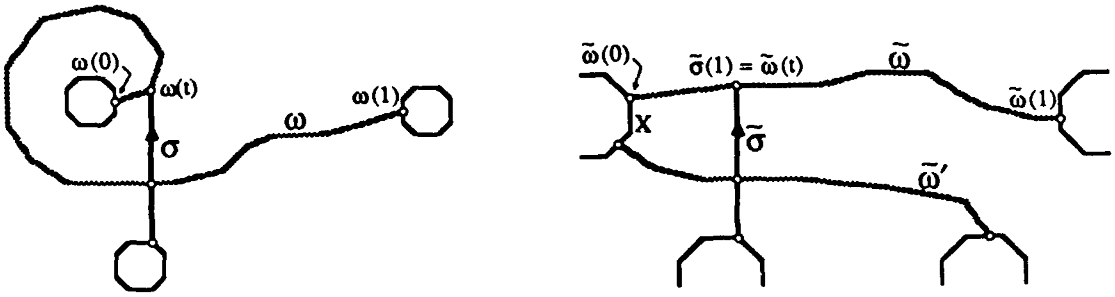


Figure 4d-4. *Lack of weak respect in a half-cut.* Even a simple half-cut can easily lack weak respect for its wire, as this example shows. At left, σ is a half-cut for ω at t. The associated cut $\sigma \star \omega_{t:0}$ is simple and therefore respects ω. But $\sigma \star \omega_{t:1}$ does not respect ω. The picture at right shows two lifts of ω that share the terminal X. One cuts a lift of $\sigma \star \omega_{t:1}$, and the other shares a terminal with it.

Lemma 4d.3. *Let η route a wire ω in the design Ω, and let σ be a half-cut for η. If σ respects ω weakly, then all cuts associated to σ have the same flow in Ω.*

Proof. Let σ be a half-cut for η at t, and let α and β be cuts associated to σ. If α and β are link-homotopic, then they automatically have the same flow, so we may assume that $\alpha \simeq_L \sigma\star\omega_{t:0}$ and $\beta \simeq_L \sigma\star\omega_{t:1}$. We must prove $\mathit{flow}(\alpha, \Omega) = \mathit{flow}(\beta, \Omega)$. It suffices to prove $\mathit{wind}(\alpha, \omega) = \mathit{wind}(\beta, \omega)$ for all wires $\omega \in \Omega$.

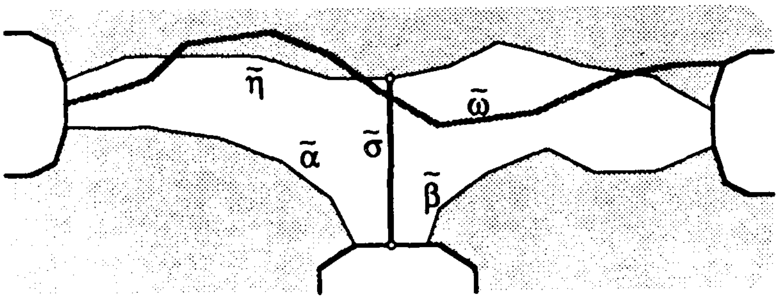


Figure 4d-5. *Flow across the associated cuts of a half-cut.* The simple links $\tilde{\alpha}$ and $\tilde{\beta}$ lift non-homotopic cuts associated to a half-cut σ for the link η, whose lift $\tilde{\eta}$ is link-homotopic to a wire lifting $\tilde{\omega}$. No wire lifting cuts $\tilde{\omega}$, so every wire lifting that cuts $\tilde{\alpha}$ or $\tilde{\beta}$ also cuts the other, unless it shares a terminal with $\tilde{\omega}$. The latter option is ruled out if σ weakly respects the design.

We imitate the proof of Proposition 4d.2. Let S be the sheet of Ω, let M be its blanket, and let $p: M \to S$ be the covering map. Lift σ and η to $\tilde{\sigma}$ and $\tilde{\eta}$ such that $\tilde{\sigma}(1) = \tilde{\eta}(t)$. Find simple links $\tilde{\alpha}$ and $\tilde{\beta}$ in M from the fringe containing $\tilde{\sigma}(0)$ to the fringes containing $\tilde{\omega}(0)$ and $\tilde{\omega}(1)$, respectively, such that $Im\,\tilde{\alpha} \cup Im\,\tilde{\beta} \cup Im\,\tilde{\eta}$ is a web of 3 threads. Then $p \circ \tilde{\alpha} \simeq_L \alpha$ and $p \circ \tilde{\beta} \simeq_L \beta$. Let $v \neq \omega$ be a wire of Ω. There are exactly $wind(\alpha, v)$ lifts of v that are cut by $\tilde{\alpha}$; because $v \neq \omega$, none of these can share a terminal with $\tilde{\eta}$. Also because they cohere with $\tilde{\eta}$, none are cut by $\tilde{\eta}$. We conclude that all such lifts are cut by $\tilde{\beta}$. Hence $wind(\alpha, v) \leq wind(\beta, v)$. By symmetry, the opposite inequality holds as well.

To establish a similar formula with ω in place of v, it suffices to show that no lift $\tilde{\omega}'$ of ω that shares a terminal with $\tilde{\eta}$ is cut by $\tilde{\alpha}$. Let $\tilde{\omega} \in [\tilde{\eta}]_L$ lift ω; then $\tilde{\omega}$ and $\tilde{\omega}'$ share a terminal. Because σ respects ω weakly, its associated cut $p \circ \tilde{\alpha}$ also respects ω weakly. Hence $\tilde{\alpha}$ cutting $\tilde{\omega}'$ would imply that $\tilde{\omega}$ could not share a terminal with $\tilde{\alpha}$. But $\tilde{\eta}$ does, and $\tilde{\eta} \simeq_L \tilde{\omega}$. We conclude that $\tilde{\alpha}$ cannot cut $\tilde{\omega}$, and hence $\tilde{\alpha}$ cannot cut $\tilde{\eta}$. Thus $wind(\alpha, v) \leq wind(\beta, v)$ even when $v = \omega$. By symmetry, this inequality is actually an equality. □

Lemma 4d.3 gives evidence that weak respect of a design is a good condition to require of a half-cut.

More bounds on flow

The technique used to prove Proposition 4d.2 and Lemma 4d.3 is a very powerful one. It compares flows by building a loop of links in a blanket and drawing correspondences among the wire liftings that cut those links. The following lemma encapsulates the technique for future use.

Lemma 4d.4. *Let γ be a simple link in a blanket M, and suppose $\alpha \simeq_P \gamma$. Every simple link in M that cuts γ either cuts some link within α, or contacts a fringe of M that intersects α but not γ.*

Proof. We inductively apply the technique of Proposition 4d.2, constructing webs in the blanket M. Let $\alpha_1, \ldots, \alpha_n$ be the links contained in α. For $1 \leq i \leq n$, let F_{i-1} be the fringe of M containing $\alpha_i(0)$, and let F_i be the fringe containing $\alpha_i(1)$. Then the terminals of $\tilde{\chi}$ are F_0 and F_n. The lemma says that every simple link in M that cuts γ either cuts some link α_i or else has some fringe F_i as a terminal where $1 \leq i < n$. To prove this claim, we construct a sequence of simple links $\kappa_0, \ldots, \kappa_{n-1}$ where κ_m has terminals F_m and F_n. At the same time, we prove by induction on m that every link that cuts γ either cuts κ_m, cuts one of the links $\alpha_1, \ldots, \alpha_m$, or has one of the fringes $F_1, \ldots, F_m$ as a terminal. Because κ_{n-1} is link-homotopic to α_n, the case $m = n-1$ will establish the claim. The basis case is easy: for $m = 0$ we let κ_0 be the simple link γ.

Now supposing the induction hypothesis is true for $m-1$, we prove it for m. It is enough to show that a simple link cutting κ_{m-1} either cuts α_m or κ_m, or else has

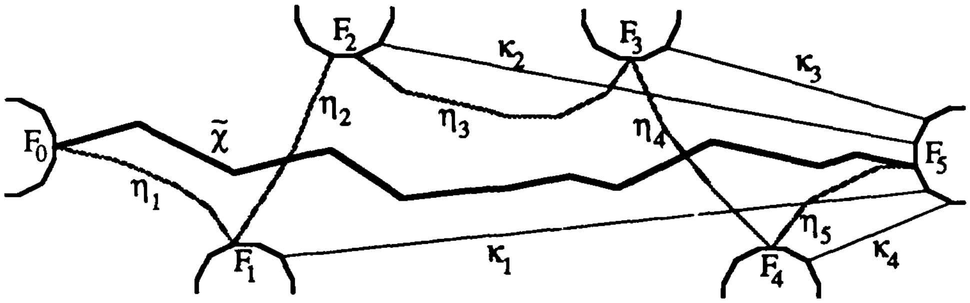


Figure 4d-6. *Flow across a chain of links.* The paths η_i are the links of a chain for $\tilde{\chi}$. By induction on m, any simple link that cuts γ either cuts κ_m, or cuts one of the links $\alpha_1, \ldots, \alpha_m$, or has one of the fringes $F_1, \ldots, F_m$ as a terminal.

F_m as a terminal. If the fringe F_m is a terminal of κ_{m-1}, then α_m is either trivial or link-homotopic to κ_{m-1}, so we can simply set $\kappa_m = \kappa_{m-1}$. Otherwise let A be the scrap of $M - Im\,\kappa_{m-1}$ that contains F_m, let $\mu_m \in [\alpha_m]_L$ be a simple link in A, and let κ_m be a simple link between F_m and F_n in the appropriate scrap of $A - Im\,\mu_m$. Then the set

$$Im\,\kappa_{m-1} \cup Im\,\mu_m \cup Im\,\kappa_m$$

is a web of 3 threads. Because its inside contains no fringes (Proposition 3b.8), every link that cuts κ_{m-1} must either cut μ_m or κ_m, or else it must have F_m as a terminal. To cut μ_m is to cut α_m. This step completes the induction, and thereby the proof. □

4E. The Branches of a Blanket

To make further progress, we need more information on degenerate links and on the lifts of a wire that contribute to the flow across a cut. We obtain the latter by studying the relation of *respect* between cuts and designs, presented in Definition 4e.1 below, which is similar to but more powerful than the relation of weak respect defined in Section 4C (Definition 4c.6). We show that every simple cut respects every design, and show that when a simple cut makes a necessary crossing with a wire in a design, the resulting *semisimple* half-cuts respect that design. Most of the cuts and half-cuts at issue in a particular design turn out to be simple or semisimple, and hence respect the design. As a by-product of this study, we discover two important correlates of nondegeneracy. First, semisimple half-cuts are nondegenerate. Second, although simple cuts can be degenerate, the ones that are have zero flow.

Degeneracy and respect are closely related: both can be best understood in terms of a division of the fringes of a blanket into *branches*. A design partitions the fringes of a blanket into branches just as it partitions the fringes of a sheet into articles. If Ω is a design on a sheet S, the articles of Ω are the components of the set $X = Bd\, S \cup \bigcup_{\omega \in \Omega} Im\, \omega$. Let M be the blanket of S, with covering map $p: M \to S$. The **branches** of the design Ω are the components of the set $p^{-1}(X)$ in M. Two different fringes in M, say A and B, are in the same branch if and only if for some wire ω in Ω, there is a sequence of fringes $A = F_0, F_1, \ldots, F_n = B$ such that for $1 \leq i \leq n$, *some* lift of ω has terminals F_{i-1} and F_i. We use the branches of a design to classify the lifts of a wire, and to identify degenerate cuts.

Degeneracy

A degenerate cut in a design Ω is one with a lifting whose endpoints fall in the same branch of Ω. For if a cut σ is degenerate in the design Ω, then σ has a chain τ in the set X. Hence by Proposition 2b.4, any lifting $\tilde{\sigma}$ of σ has a path-homotopic lifting $\tilde{\tau}$ of τ which lies in $p^{-1}(X)$. Conversely, if $\tilde{\sigma}$ is any lift of σ whose endpoints lie in the same branch, then there is path $\tilde{\tau}$ in $p^{-1}(X)$ between the endpoints of $\tilde{\sigma}$. (For some wire $\omega \in \Omega$, it is the concatenation of subpaths of lifts of ω with paths along fringes.) By Lemma 2a.5, there is a path homotopy F between $\tilde{\sigma}$ and $\tilde{\tau}$, and the projection of F is a path homotopy between σ and a path $\tau: I \to X$.

This characterization of degeneracy clarifies several facts about degenerate cuts. First, a cut that is degenerate in one design is also degenerate in any embedding of that design. Second, *if one associated cut of a subcut is degenerate*, then all are. Third, the concatenation of degenerate subcuts is degenerate.

Strong respect

For a cut to respect a design strongly means, in essence, that each branch of the design contributes at most one necessary crossing to the flow across the cut.

Definition 4e.1. Let Ω be a design on a sheet S. A cut χ of S **respects** Ω (strongly) if whenever

(a) ω is a wire in Ω,

(b) $\tilde{\omega}$ and $\tilde{\omega}'$ are two lifts of ω in the same branch of Ω, and

(c) $\tilde{\chi}$ is a lift of χ that cuts $\tilde{\omega}$,

the terminals of $\tilde{\omega}'$ lie within the same side of $\tilde{\chi}$.

The definition of strong respect (4e.1) differs from that of weak respect (4c.6) only in part (b), where the requirement that ω and ω' lie in the same branch of Ω has replaced the weaker condition that ω and ω' share a terminal. Like weak respect, strong respect is invariant under link homotopy.

The significance of Definition 4e.1 will become increasingly clear in later sections. From a technical point of view, it is central to the study of the design model. In the remainder of this section, we give sufficient conditions for a cut to respect a design.

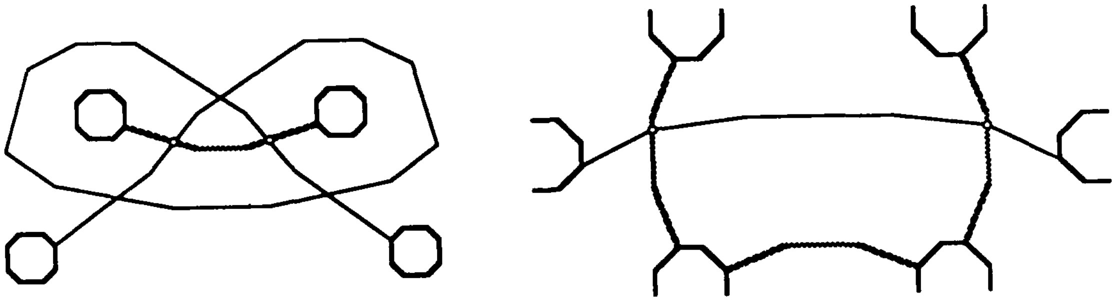


Figure 4e-1. *Weak respect but not strong respect.* The pretzel-shaped cut at left respects the wire shown in grey weakly but not strongly. At right are the relevant liftings. The three wire liftings (grey) lie in the same branch of the design. Two of them cut the cut lifting (thin black path), but they do not share a terminal.

Figure 4e-1 shows a nearly minimal example of a cut that weakly respects a design without strongly respecting it. As the next result shows, in any such example the cut must have self-intersections.

,***Proposition 4e.2.*** *Simple cuts respect all designs.*

Proof. Let χ be a simple cut of a sheet S, and let ω be a wire in S. Suppose $\tilde{\chi}$ and $\tilde{\omega}$ are lifts of χ and ω that cut one another. For $n \geq 1$, we say $\tilde{\chi}$ ***n*-respects** ω if whenever $\alpha 0n$ are distinct lifts of ω starting with $\tilde{\omega}$ such that for $1 \leq i \leq n$, the links α_{i-1} and α_i share a terminal, the terminals of α_n lie on the same side of $\tilde{\chi}$. Then 1-respect is the same as weak respect, and strong respect is n-respect for all n. We prove by induction on n that χ n-respects ω. The basis case is established by Lemma 4c.7; it says that χ respects Ω weakly because χ is simple.

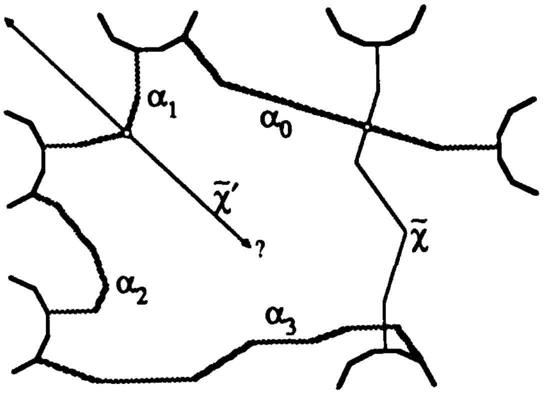


Figure 4e-2. *Why simple cuts respect all designs strongly.* In the situation depicted, a simple cut χ with lifting $\tilde{\chi}$ does not n-respect a wire ω. (Here $n = 3$.) A branch of ω includes liftings $\alpha_0, \ldots, \alpha_n$, of which α_0 cuts $\tilde{\chi}$, and α_n cuts or shares a terminal with $\tilde{\chi}$. The link $\tilde{\chi}'$ is to α_1 as $\tilde{\chi}$ is to α_0. Where can its terminals lie? Wherever $\tilde{\chi}'$ goes it contradicts the induction hypothesis that χ $(n-1)$-respects ω.

For the induction step, where $n \geq 2$, we suppose that α_n either cuts or shares a terminal with $\tilde{\chi}$, and derive a contradiction. By the induction hypothesis, none of the links α_i for $i < n$ intersects $\tilde{\chi}$. Furthermore, the lifts of ω are all disjoint since

ω is simple. Hence $\tilde{\chi}$ and $\alpha_1, \ldots, \alpha_n$ form a web of $n-1$ threads (if α_n cuts $\tilde{\chi}$) or n threads (if it does not), as shown in Figure 4e-2. Let γ be a simple link whose image is the thread of this web that intersects $Im\,\tilde{\chi}$. Let h be a covering transformation that takes α_0 to α_1, and put $\tilde{\chi}' = h \circ \tilde{\chi}$. Then $\tilde{\chi}'$ cuts α_1, because $\tilde{\chi}$ cuts α_0.

Consider how $\tilde{\chi}$ leaves the inside of the web. One terminal of $\tilde{\chi}'$ lies on the other side of α_1 from $\tilde{\chi}$. Regarding the position of the other terminal, there are three possibilities: it may be a terminal of α_i for some $i \neq 1$, or $\tilde{\chi}'$ may cut α_i for some $i \neq 1$, or $\tilde{\chi}'$ may cut γ. Since $\tilde{\chi}'$ does not cut $\tilde{\chi}$, as these are lifts of a simple link, one or two applications of Lemma 4c.1 show that if $\tilde{\chi}'$ cuts γ, it also cuts α_i where i is either 0 or n. In each case, apply the induction hypothesis to the lift $\tilde{\chi}'$ of χ, and the lifts α_1 and α_i of ω. Because $\tilde{\chi}$ cuts α_1, and χ k-respects ω where $k = |i-1|$, the link α_i cannot cut $\tilde{\chi}'$ or share a terminal with it. This contradiction means that χ n-respects ω, completing the induction. □

One interesting consequence of Proposition 4e.2 is the following.

Corollary 4e.3. *Degenerate simple cuts have zero flow.*

Proof. More precisely, if χ is a simple cut that is degenerate in a design Ω, then $\mathit{flow}(\chi, \Omega) = 0$. For if χ is degenerate in Ω, then any lift $\tilde{\chi}$ of χ is path-homotopic to a path β in a single branch B of Ω. If the article C that is the projection of B is a single fringe, then so is B, which makes χ trivial. In this case no simple link cuts $\tilde{\chi}$, and so $\mathit{flow}(\chi, \Omega) = 0$. So suppose C contains a wire $\omega \in \Omega$. By Lemma 4d.4, no lift of a wire in Ω can cut $\tilde{\chi}$ unless it intersects a fringe of β or cuts a link in β. In either case it intersects B, and is therefore part of B, and hence is a lift of ω. But the terminals of $\tilde{\chi}$ lie in B, and by Proposition 4e.2, χ respects ω. Therefore no lift of ω in B cuts $\tilde{\chi}$. We conclude that no wire in Ω has a lifting that cuts $\tilde{\chi}$, and so $\mathit{flow}(\chi, \Omega) = 0$. □

Half-cuts and mid-cuts

As one might expect, we say that a half-cut or mid-cut respects a design if and only if its associated cuts do. Because respect is invariant under link homotopy, half-cuts and mid-cuts that are akin respect the same designs. Our next result gives us a tool for constructing respectful subcuts from respectful cuts. With Lemma 4e.4 and Proposition 4e.2 together, we have enough leverage to prove that almost any useful subcut respects its design.

Lemma 4e.4. *Let Ω be a design on the sheet S, let ω be a routing of a wire in Ω, and let χ be a cut of S that respects Ω. If (s,t) is a necessary crossing of χ by ω, then for $e \in \{0,1\}$, every cut in $[\chi_{0:s} \star \omega_{t:e}]_L$ respects Ω.*

Proof. Let M be the blanket of S, with covering map $p\colon M \to S$. Because respect is invariant under link homotopy, no generality is lost by assuming that $\omega \in \Omega$. So let $\tilde{\chi}$ and $\tilde{\omega}$ be lifts of χ and ω such that $\tilde{\chi}(s) = \tilde{\omega}(t)$, and let β be any simple link in $[\tilde{\chi}_{0:s} \star \tilde{\omega}_{t:e}]_L$. Also let v be any wire in Ω. It suffices to show that for any lift of v that cuts β, every other lift of v in the same branch has its terminals in the same scrap of β.

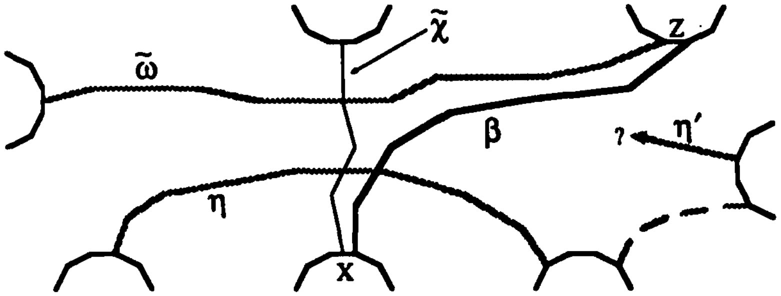


Figure 4e-3. *Construction of respectful subcuts.* The lift $\tilde{\chi}$ of χ cuts the lift $\tilde{\omega}$ of the link ω, and χ respects the design. No wire lifting cuts $\tilde{\omega}$, so any wire lifting that cuts β also cuts $\tilde{\chi}$. Of any two wire liftings in the same branch, at most one can cut β. We show that the other cannot even share a terminal with β.

I claim that if a lift $\tilde{v}$ of v cuts β, then it also cuts $\tilde{\chi}$. The link $\tilde{\chi}$ cuts $\tilde{\omega}$ because the crossing (s,t) is necessary. The link $\tilde{v}$ does not cut $\tilde{\omega}$, because their projections to S form a subdesign of Ω. In other words, $\tilde{v}$ coheres with $\tilde{\omega}$. Now apply Lemma 4c.1: because $\tilde{v}$ cuts β but not $\tilde{\omega}$, it must cut $\tilde{\chi}$.

Let $\tilde{v}$ and $\tilde{v}'$ be two lifts of v in the same branch of Ω, and suppose $\tilde{v}$ cuts β. Then $\tilde{v}$ cuts $\tilde{\chi}$, and because χ respects Ω, the other lift $\tilde{v}'$ neither cuts $\tilde{\chi}$ nor shares a terminal with it. In particular, $\tilde{v}'$ does not cut β, and it does not share a terminal with $\beta(0)$, because $\beta(0)$ lies on a terminal of $\tilde{\chi}$. One other possibility remains: that $\tilde{v}'$ might share a terminal with $\beta(1)$. But $\beta(1)$ lies on a terminal of $\tilde{\omega}$, so if $\tilde{v}'$ shares this terminal, then $\tilde{\omega}$ and $\tilde{v}$ lie in the same branch of Ω. Both these links cut $\tilde{\chi}$, however, and since χ respects Ω, they cannot lie in the same branch. We conclude that if $\tilde{v}$ cuts β, then the terminals of $\tilde{v}'$ lie on the same side of β. Therefore β respects Ω. □

To put Lemma 4e.4 into practice, we define an important class of half-cuts to which it applies.

Definition 4e.5. Let ω route a wire in the design Ω. A half-cut σ for a link ω is **semisimple** in Ω if there is a simple cut χ and a necessary crossing (c,t) of χ by ω such that $\chi_{0:c}$, as a half-cut for ω at t, is akin to σ.

One could extend Definition 4e.5, and the following proposition as well, to mid-cuts. A mid-cut τ between links v and ω would be semisimple if there were a simple cut χ and necessary crossings (a,s) and (b,t) of χ by v and ω, respectively, such that $\chi_{a:b}$, as a mid-cut between v at s and ω at t, was akin to τ. As natural as the generalization is, I have no use for it.

Proposition 4e.6. *Semisimple half-cuts are nondegenerate and respectful.*

Proof. Let σ be a semisimple half-cut in the design Ω. The claim is that σ respects Ω and is nondegenerate iu Ω. Let χ be the simple cut whose half-cut $\chi_{0:c}$ is akin to σ. Then χ respects Ω by Proposition 4e.2, and by Lemma 4e.4, every cut associated to $\chi_{0:c}$ respects Ω. Thus $\chi_{0:c}$ respects Ω. Since half-cuts that are akin respect the same designs, σ also respects Ω.

To see that σ, or equivalently $\chi_{0:c}$, is nondegenerate, let $\tilde{\chi}$ and $\tilde{\omega}$ be lifts of $\tilde{\chi}$ satisfying $\tilde{\chi}(c) = \tilde{\omega}(t)$. If $\chi_{0:c}$ were degenerate, then the endpoints of $\tilde{\chi}_{0:c} \star \tilde{\omega}_{t:1}$ would lie in the same branch of Ω. And since ω routes a wire v in Ω, this branch would include a lifting $\tilde{v} \in [\tilde{\omega}]_L$ cutting $\tilde{\chi}$, as well as the fringe containing $\tilde{\chi}(0)$. But χ respects Ω, so this cannot happen. □

4F. Safety of Cuts and Half-Cuts

So far we have only considered the topology of designs, using concepts such as necessary crossings and the flows across cuts. Now we mix in some geometry: the arc lengths and capacities of cuts. The result is a powerful set of lemmas concerning the safety of cuts. For example, Corollary 4f.5 shows that an unsafe, major, simple cut gives rise to an unsafe, major, ***straight*** cut, and therefore every major simple cut in a safe design is safe..

The technique we use generalizes that discovered by Cole and Siegel [6] and independently by Leiserson and the author. It involves shrinking the cut to its elastic chain, and relating the flow and capacity of the cut to the flows and capacities of the links in the elastic chain. If the cut respects its design, the flow across the chain can be smaller than the flow across the cut only by the widths of the wires whose terminals touch the middle of the chain. But in going from the cut to the chain, the total capacity is reduced by the width of every fringe that touches the middle of the chain. Since wires are no wider than their terminals, the total capacity decreases by at least as much as the total flow. Hence if the cut was unsafe, one of the links in the chain is unsafe as well. Of course, I have glossed over some technical issues, such as the need to ignore minor cuts.

Redefinition of safety

In order to accommodate half-cuts as well as cuts, we redefine safety in terms of flow. Let θ be any cut or half-cut. The **margin** of θ in a design Ω is the capacity of θ minus the flow across θ:

$$margin(\theta, \Omega) = cap(\theta, \Omega) - flow(\theta, \Omega).$$

The terminology is meant to suggest 'margin of safety'. We say θ is **unsafe** in the design Ω if and only if $margin(\theta, \Omega)$ is negative. A cut whose margin is 0 is called **marginal** (or "marginally safe").

Chains of links

Proposition 4f.1, which forms the basis for the results of this section, relates the flow across a cut or half-cut to the flow across a chain of links and half-links. Recall from Chapter 3 that a **chain** for a path in a manifold is any homotopic path, and from Section 3D that every path in a sheet has a unique elastic chain.

A chain for a cut or half-cut consists of *major links* and *gaps*, which are defined as follows. Let α be a chain for a cut or half-cut σ in a design Ω. If σ is a half-cut, then α ends with a half-link τ. Aside from this, α consists of major links α_1, $\ldots$, α_n interspersed with minor links and paths along fringes. The **major links** of the chain α are the paths α_i, plus τ if τ is nondegenerate. (If τ is degenerate, we say the chain is degenerate.) The portions of α between its major links are called the **gaps** in α. Each gap γ can intersect at most one article C of Ω. For a gap consists of empty and degenerate links, and none of these connect different articles of a design. The **width** of the gap γ is defined to be the width of the wire in the article C, or zero if C includes no wire.

The **flow** across a chain is just the sum of the flows across its major links, but the capacity of a chain is slightly more complex. We denote by $gaps(\alpha)$ the sum of the widths of the gaps in a chain α. If α is a chain for χ, the quantity $gaps(\alpha)$ represents the amount of wiring that might contribute to the flow across χ but escape detection by any link of χ. The **capacity** of a chain is the sum of the capacities of its major links, plus the sum of the widths of its gaps.

The flow across a chain

We use three tools to derive sufficient conditions for the existence of an unsafe or marginal link in a chain. One is Proposition 4f.1, coming up, which bounds from below the flow across a chain. Another is Lemma 4f.2, which gives conditions for the existence of a link in the chain. The third is Lemma 4f.3, which bounds from above the capacity of a chain.

Proposition 4f.1. *Let σ be a cut or half-cut that respects the design Ω. If α is a chain of straight links for σ, then*

$$flow(\alpha, \Omega) \geq flow(\sigma, \Omega) - gaps(\alpha). \qquad (4\text{–}3)$$

Proof. The first step is to reduce the case of a half-cut to that of a cut. If σ is a cut, put $\eta = \alpha$ and $\chi = \sigma$. Now suppose σ is a half-cut for a wire ω at t. Let χ be a cut in $[\sigma \star \omega_{t:1}]_P$, and let η be the path $\alpha \star \omega_{t:1}$. Then η is a chain for χ. The cut χ respects Ω because σ respects Ω, and χ is associated to σ. By Proposition 4b.3 and the definition of the flow across a half-cut, we have $\mathit{flow}(\chi, \Omega) = \mathit{flow}(\sigma, \Omega)$. We also have $\mathit{flow}(\eta, \Omega) = \mathit{flow}(\alpha, \Omega)$, because their nondegenerate links correspond. (Empty links are irrelevant because they have zero flow.) In other words, the only way η and α differ is that where α has a half-link, η has an associated cut for that half-link. One is degenerate if and only if the other is degenerate. Hence it suffices to prove

$$\mathit{flow}(\eta, \Omega) \geq \mathit{flow}(\chi, \Omega) - \mathit{gaps}(\alpha). \tag{4–4}$$

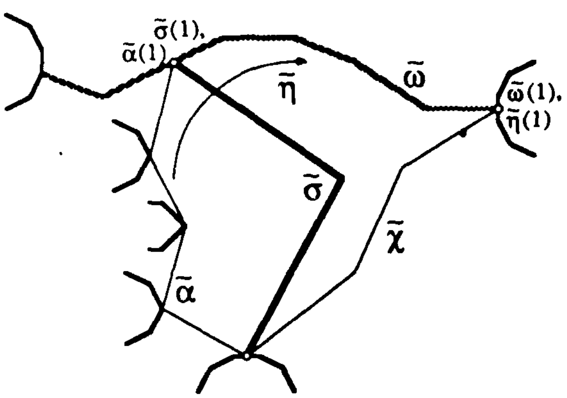


Figure 4f-1. *Bounding the flow across a chain.* This figure depicts liftings of certain paths and chains involved in Proposition 4f.1. Here σ is a half-cut for ω. Its chain α reaches only to $\sigma(1)$, so we replace σ by its associated cut χ, and α by a chain η for χ. Though the links of α are straight, the final link of η need not be.

We apply Lemma 4d.4 to lifts of χ and η for each wire v in Ω. Using Proposition 2b.4, let $\tilde{\chi}$ and $\tilde{\eta}$ be path-homotopic lifts of χ and η, respectively. Let η_1, $\ldots$, η_m be the links of η, and let $\tilde{\eta}_1$, $\ldots$, $\tilde{\eta}_m$ be the corresponding subpaths of $\tilde{\eta}$. By Lemma 4d.4, any lift of v that cuts $\tilde{\chi}$ either cuts $\tilde{\eta}_i$ for some i, or else has as terminal a fringe that intersects $\tilde{\eta}$ but not $\tilde{\chi}$. Let B denote the branch of v that contains this fringe. Because χ respects Ω, the branch B contains at most one lift of $\tilde{v}$ that cuts $\tilde{\chi}$, and B does not include either terminal of $\tilde{\chi}$.

The contribution of B to $\mathit{flow}(\chi, \Omega)$ can be charged to a gap in α of width $\mathit{width}(v)$ in such a way that no gap is charged twice. Let M be the blanket, and let $\tilde{\alpha}$ be the lift of α to M such that $\tilde{\alpha}(0) = \tilde{\eta}(0) = \tilde{\chi}(0)$. Pick $x \in (0,1)$ such that $\tilde{\alpha}(x) \in B \cap \mathit{Bd}\, M$, and let $[s,t]$ be the maximal interval containing x such that $\gamma = \alpha_{s:t}$ consists of minor links interspersed with paths along fringes. If α ends with a degenerate half-cut, this may be included in γ. The only fringes that γ touches are those in the article of Ω that includes $\mathit{Im}\, v$. Let $\tilde{\gamma}$ be the subpath of $\tilde{\eta}$ corresponding to γ.

To show that γ is a gap of α, it is enough to prove $s \neq 0$ and $t \neq 1$. The links (and possible terminating half-link) of γ are straight and minor, and since the terminals of v are convex, each link is nonempty and therefore degenerate instead. All the fringes that intersect $\tilde{\gamma}$ are part of the same branch B. So if $s = 0$, then B contains $\tilde{\gamma}(0) = \tilde{\chi}(0)$. Or if $t = 1$, then the final link or half-link of α is degenerate, and B contains $\tilde{\chi}(1)$. But B includes neither terminal of $\tilde{\chi}$, so $s > 0$ and $t < 1$. Thus γ is a gap of α, and its width is $width(v)$. We charge the contribution of B to γ. Because all the fringes that $\tilde{\gamma}$ touches are part of B, and B contains at most one lift of $\tilde{v}$ that cuts $\tilde{\chi}$, the gap γ is charged only once.

The upshot of this analysis is that the difference between the flow across χ and the flows across the links η_i is accounted for by the widths of the gaps in α. In symbols,

$$flow(\chi, \Omega) - \sum_{i=1}^{m} flow(\eta_i, \Omega) \leq gaps(\alpha).$$

To prove inequality (4–5), it remains to identify $flow(\eta, \Omega)$, the sum of the flows across the major links of η, with $\sum_{i=1}^{m} flow(\eta_i, \Omega)$. In other words, we must show that the minor links of η contribute nothing to the flow across χ. Certainly the empty ones contribute nothing, because their flow is zero in Ω. The straight degenerate links, also, have zero flow in Ω by Lemma 4e.3. The remaining case is the final link η_m of η, which can be nonsimple if σ is a half-cut. If η_m is degenerate, then the endpoints of $\tilde{\eta}_m$ lie in the same branch T of Ω. Thus $\tilde{\eta}_m$ is path-homotopic to a path in a single branch T of Ω, and hence it cannot cut links in any other branch. The branch T contains a terminal of $\tilde{\chi}$, because $\tilde{\eta}_m(1)$ and $\tilde{\chi}(1)$ lie on the same fringe. Because χ respects Ω, no link in T that lifts a wire in Ω can cut $\tilde{\chi}$. Therefore no lift of a wire in Ω can cut both $\tilde{\eta}_m$ and $\tilde{\chi}$. In other words, η_m contributes nothing to the flow across χ. □

Elastic chains

As mentioned before, our strategy is to reduce a major cut or nondegenerate half-cut to the major links in its elastic chain. The next lemma ensures that some major link always remains. It uses the fact that all the links and half-links in an elastic chain are straight (Lemma 3d.5).

Lemma 4f.2. *The elastic chain for a major cut or a nondegenerate half-cut includes at least one major link.*

Proof. First consider the case of a cut. Let χ be a major cut in the design Ω, and let α be the elastic chain for χ. If the endpoints of χ lie in different articles of Ω, then α must contain a link that passes between two different articles, and this link is major. So we assume that all the links in α have their endpoints in the same article. Suppose first that this article contains no wire of Ω. Then $gaps(\alpha) = 0$, but since χ is nonempty, its flow is positive. Hence by Proposition 4f.1, the flows across the major links of α sum to a positive quantity, and therefore α must have a major link. Now suppose the article contains a wire ω of Ω. Using Proposition 2b.4, lift χ and α to path-homotopic paths $\tilde{\chi}$ and $\tilde{\alpha}$. The endpoints of $\tilde{\chi}$ lie in different branches of ω because χ is nondegenerate. Hence $\tilde{\alpha}$ contains a link that passes between two branches of ω, and consequently α has a nondegenerate link β. Since β is straight and terminals are convex, β has two terminals. Thus β is a nonempty, nondegenerate link of α.

Now consider the case of a half-cut. Let σ be a nondegenerate half-cut for a link ω at t, and let η be the elastic chain for σ. Let χ be a cut in $[\sigma \star \omega_{t:1}]_P$, and let α be the chain $\eta \star \omega_{t:1}$ for χ. The cut χ is nondegenerate in Ω because σ is nondegenerate in Ω and χ is associated to σ. As before, we may assume that links of α all lie in the same article of Ω. This article contains a wire in Ω, of which ω is a route. Let $\tilde{\chi}$ and $\tilde{\alpha}$ be path-homotopic lifts of χ and α. Then the endpoints of $\tilde{\chi}$ lie in different branches of ω. Hence $\tilde{\alpha}$ contains a link that passes between two branches of ω, and consequently α has a nondegenerate link β. If β is part of η, then β is straight, and hence nonempty as shown above. Otherwise β is an associated cut of the final half-cut in η. Since β is nondegenerate, this half-cut is nondegenerate, and again η includes a major link. □

The most important fact about elastic chains is that they have minimal euclidean arc length, and minimal arc length in all other norms as well (Corollary 3d.8). Combined with the following lemma, this fact implies that the capacity of a cut or half-cut is no smaller than the capacity of its elastic chain.

Lemma 4f.3. *The elastic chain θ of a major cut or nondegenerate half-cut σ satisfies $cap(\theta) - \|\theta\| \leq cap(\sigma) - \|\sigma\|$, and the inequality is strict if θ is degenerate.*

Proof. We assume, of course, that the widths of wires and fringes are specified by some design Ω. Let $\theta_1, \ldots, \theta_n$ be the major links of θ, where $n \geq 1$ by Lemma 4f.2. For $1 \leq i < n$ the points $\theta_i(1)$ and $\theta_{i+1}(0)$ lie in the same article C_i of Ω. Let C_0 denote the article containing $\theta_1(0)$, and let C_n be the article containing $\theta_n(1)$. For $1 \leq i \leq n$, let a_i be the width of the detail containing $\theta_i(0)$, and let b_i be the width of the detail containing $\theta_i(1)$. Also put w_i equal the width of the wire in the article C_i, if any, and otherwise put $w_i = 0$. Then $\sum_{i=1}^{n-1} w_i = gaps(\theta)$, and hence by the definition of the capacity of θ we have

$$cap(\theta) = \sum_{i=1}^{n} cap(\theta_i) + \sum_{i=1}^{n-1} w_i. \tag{4-5}$$

No wire is wider than either of its terminals, and therefore $w_i \leq a_i$ and $w_i \leq b_i$ for

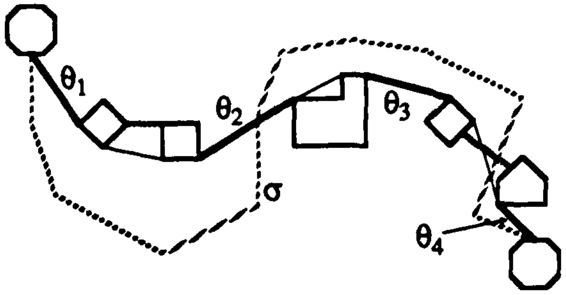


Figure 4f-2. *A simple cut and its elastic chain.* Here σ is a cut in a design whose wires are shown in grey. The elastic chain θ for σ has three minor links (thin black lines) and four major links $\theta_1, \ldots, \theta_4$ (thick black lines).

each i.

Now we evaluate inequality (4–3) using the definition of capacity for cuts:

$$
\begin{aligned}
cap(\theta) &= \sum_{i=1}^{n}\Big(\|\theta_i\| - a_i/2 - b_i/2\Big) + \sum_{i=1}^{n-1} w_i \\
&\leq \left(\sum_{i=1}^{n}\|\theta_i\|\right) - a_1/2 - b_n/2 \\
&\leq \|\theta\| - a_1/2 - b_n/2. \qquad (4\text{–}6)
\end{aligned}
$$

The final inequality holds because the paths θ_i are disjoint subpaths of θ, and hence is strict if θ contains anything but major links. In particular, it is strict if θ is degenerate. Comparing inequality (4–6) to $cap(\sigma)$, which is $\|\sigma\| - a_1/2 - b_n/2$, we get the inequality

$$cap(\theta) - \|\theta\| \leq -a_1/2 - b/2 \leq cap(\sigma) - \|\sigma\|\,,$$

with strictness if θ is degenerate. □

Applications

There are three main applications of Proposition 4f.1 and Lemmas 4f.2 and 4f.3. One concerns unsafe cuts, one concerns unsafe half-cuts, and the third concerns half-cuts of margin zero. The first two are given here; the third is discussed in Section 5B. All the applications start with a cut or half-cut, compare the flow and capacity of an elastic chain, and show that one of the links in that chain must have low margin.

Lemma 4f.4. *Let χ be a major simple cut in a design Ω. There is a major straight cut β satisfying $margin(\beta, \Omega) \leq margin(\chi, \Omega)/n$ for some $n > 0$.*

Proof. Let α be the elastic chain for χ. By Lemma 4f.3 and Corollary 3d.8, the chain α satisfies $cap(\alpha) \leq cap(\chi)$. Let $\alpha_1, \ldots, \alpha_n$ be the major links of α. Because

χ is simple, it respects Ω, by Proposition 4e.2. Hence Proposition 4f.1 applies to χ, α, and Ω. The result is

$$\sum_{i=1}^{n} \mathit{flow}(\alpha_i, \Omega) \geq \mathit{flow}(\chi, \Omega) - \mathit{gaps}(\alpha).$$

The definition of $\mathit{cap}(\alpha)$ and the fact that $\mathit{cap}(\alpha) \leq \mathit{cap}(\chi)$ imply

$$\sum_{i=1}^{n} \mathit{cap}(\alpha_i, \Omega) \leq \mathit{cap}(\chi) - \mathit{gaps}(\alpha).$$

Subtracting the previous inequality from this one shows that $\sum_{i=1}^{n} \mathit{margin}(\alpha_i, \Omega) \leq \mathit{margin}(\chi, \Omega)$. Because $n > 1$, by Lemma 4f.2, there must be some link β among the α_i such that $\mathit{margin}(\beta, \Omega) \leq \mathit{margin}(\chi, \Omega)/n$. This link β is a major straight cut. □

If one applies Lemma 4f.4 to a major simple cut with negative margin, the major straight cut it produces has negative margin, and is therefore unsafe. In a safe design, this cannot happen.

Corollary 4f.5. *All major simple cuts in a safe design are safe.* □

The second application concerns half-cuts. Its proof is very similar to that of Lemma 4f.4, except for some additional concern about degeneracy.

Lemma 4f.6. *Let ω route a wire in a safe design Ω. If ω has an unsafe, nondegenerate, simple half-cut that respects Ω, then ω has an unsafe, nondegenerate, straight half-cut.*

Proof. Let σ be the unsafe, nondegenerate, simple half-cut for ω, and let α be the unique elastic chain for σ. Let $\alpha_1, \ldots, \alpha_n$ be the major links of α. Lemma 4f.2 and Corollary 3d.8 imply $\mathit{cap}(\alpha) \leq \mathit{cap}(\sigma)$, and Lemma 3d.5 implies that the links of α are straight. Because σ respects Ω, Proposition 4f.1 shows that

$$\sum_{i=1}^{n} \mathit{flow}(\alpha_i, \Omega) \geq \mathit{flow}(\sigma, \Omega) - \mathit{gaps}(\alpha).$$

By the definition of $\mathit{cap}(\alpha)$ and the fact that $\mathit{cap}(\alpha) \leq \mathit{cap}(\sigma)$, we also have

$$\sum_{i=1}^{n} \mathit{cap}(\alpha_i, \Omega) \leq \mathit{cap}(\sigma) - \mathit{gaps}(\alpha).$$

Subtracting the previous inequality from this one shows that

$$\sum_{i=1}^{n} margin(\alpha_i, \Omega) \leq margin(\sigma, \Omega). \qquad (4\text{–}7)$$

The right-hand side of (4–7) is negative by assumption. Since Ω is safe, every major straight cut in Ω has nonnegative margin. Hence not every α_i can be a link; α_n must be a nondegenerate half-cut τ for ω. Then inequality (4–7) implies $margin(\tau, \Omega) \leq margin(\sigma, \Omega) - \sum_{i=1}^{n-1} margin(\alpha_i, \Omega)$, which means $margin(\tau, \Omega) < 0$. Therefore τ is an unsafe, nondegenerate, straight half-cut for ω. □

Chapter 5

Routing a Safe Design

The most natural way to prove that a safe design has a proper embedding is to construct one. And, in fact, all the methods I have considered for proving the design routability theorem are essentially constructive. But the construction can tend toward either of two extremes. It might be a deterministic algorithm that builds the embedding step by step, maintaining some invariant that ensures that the final embedding is proper. Or it might be a mathematical description that distinguishes a certain design; the description could involve limits and other infinitary "constructors". One would then need to prove the existence of the limits, and deduce from the description that the resulting design is proper.

Many methods for proving the design and sketch routability theorems succeed without being particularly enlightening. One algorithmic approach, for example, is to begin with rubber bands and slowly move them apart, keeping them taut and bending them as necessary, until they reach their natural width. This process gives rise to a constructive proof of the sketch routability theorem and a routing algorithm that runs in time $\Theta(n^7)$ or so [33]. The mathematical approach, also, can probably be made to work. One can prove theorems similar to the sketch routability theorem in the grid-based wiring model, as Cole and Siegel claimed in [6]. Letting the grid size approach zero and taking the limit of embeddings with minimal wire length, one can probably obtain proper embeddings for safe sketches in other wiring models as well. (There are some difficult technical issues concerning self-avoidance.) But again, this approach gives little guidance for developing an efficient routing algorithm and proving it correct.

My construction, presented in this chapter, is a compromise that lies closer to the mathematical extreme. Given a safe design, we first construct an *evasive* route for each wire. We then prove that each wire in the design has a minimum-length evasive route. (Here a limiting process comes in, via Proposition 2c.8.) Finally, we characterize these routes in sufficient detail to show that they form a proper design. Thus we prove the hard direction of the design routability theorem, and with some extra work in Chapter 6, we get the design routing theorem as well. But the real advantage of this approach shows up in Chapter 7: the information it provides about ideal routes allows us to develop efficient algorithms for constructing them.

Inspiration

The idea of using minimum-length evasive routes for single-layer routing first appeared in a paper by Tompa [52], who considers river routing across a channel. I have adapted his construction to the case of a multiply connected routing region, but the outline of the proof is the same. Unfortunately, the technical difficulties of working in a blanket, rather than in the plane, make my proof about fifteen times longer than his.

A simplified problem

What follows is a brief overview of Tompa's construction. I have simplified it by using a piecewise linear wiring norm rather than the euclidean norm used in [52]. Suppose one wishes to connect terminals $a_1, \ldots, a_n$ on the bottom of a rectangular channel to terminals $b_1, \ldots, b_n$ on the top, using "wires" of unit width. Assume these terminals are numbered from left to right. We argue that the conditions

$$\left\{ \begin{array}{l} \|a_i - a_j\| \geq |i-j| \\ \|a_i - b_j\| \geq |i-j| \;\Big|\; 1 \leq i,j \leq n \\ \|b_i - b_j\| \geq |i-j| \end{array} \right\}, \qquad (*)$$

are necessary and sufficient for the channel to be routable by wires that remain at least one unit apart. We can interpret these inequalities as saying that certain nondegenerate cuts are safe. If $c_i \in \{a_i, b_i\}$ and $c_j \in \{a_j, b_j\}$, then the "capacity" of the "cut" from c_i to c_j is $\|c_i - c_j\| - 1$; the "flow" across this cut is $\max\{0, |i-j|-1\}$; and if $i = j$, then the cut from c_i to c_j is "degenerate". See Figure 5-1(i).

Suppose that one of the inequalities in (*) fails to hold. Then for some c_i and c_j, with $i \neq j$, the line segment χ from c_i to c_j has length less than $|i-j|$. No matter how the wires are routed, each of the wires whose index lies between i and j must cross χ. Counting also the wires i and j, which intersect the endpoints of χ, there are $|i-j|+1$ different wires that must intersect χ. They cannot do so and still remain one unit apart. Therefore condition (*) is necessary.

To show that condition (*) is sufficient, we route the wires assuming that it holds. From the perspective of wire i, each terminal $c_j \in \{a_j, b_j\}$ for $j \neq i$ presents a *barrier* for wire i. The barrier surrounding c_j is the set $\{\, x : \|x - a_j\| < |i-j| \,\}$. We call this a *left* barrier if $j < i$, and a *right* barrier if $j > i$. Part (ii) of Figure 5-1 suggests why left and right barriers do not intersect. If the barrier around c_j with $j < i$ intersected the barrier around c_k with $k > i$, there would be a point x such that $\|x - c_j\| < |i-j|$ and $\|x - c_k\| < |i-k|$. Then the triangle inequality would

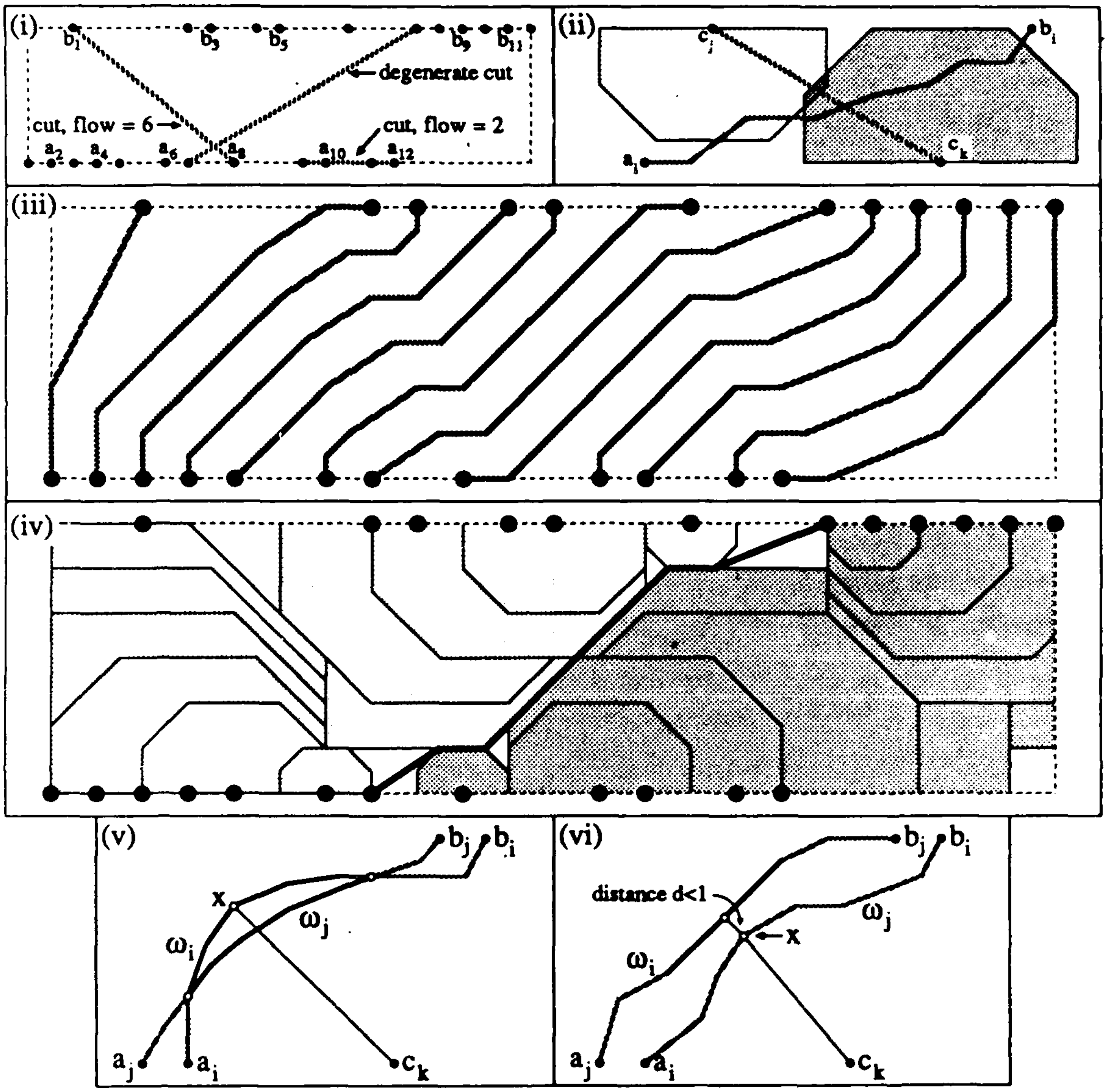


Figure 5-1. *A simplified routing problem.* The chief ideas behind the routing of safe designs are amply illustrated by the much simpler problem of river routing in a simply connected region. This problem supports analogues of the design routability and routing theorems. Under natural definitions of safety and degeneracy for cuts, shown in part (i), the terminals of a channel can be connected by wires of unit width if and only if every nondegenerate cut is safe. Furthermore, if the channel can be routed, one can route it using minimum-length *evasive* wires, as shown in part (iii). An evasive wire is one that avoids the *barriers* presented by the terminals of other wires (part (iv)). The remaining parts illustrate the argument that the minimum-length evasive wires exist and are sufficiently separated.

imply $\|c_j - c_k\| < |j - k|$, contradicting $(*)$. Hence there is a path from a_i to b_i that avoids both its left and right barriers. Such a path is called an *evasive* route for wire i. We route wire i by choosing the minimum-length evasive path ω_i from a_i to b_i, as shown in part (iii) of Figure 5-1. Part (iv) of that figure illustrates the barriers for a particular wire.

Now for the interesting part: we prove, in three steps, that the minimum-length evasive routes ω_i stay one unit apart. The first step appeals to the reader's geometric intuition, although a rigorous proof could be provided. Because the wiring norm is piecewise linear, the barriers are polygonal. The first claim is that the routes are piecewise straight. We also claim that wherever ω_i has a joint t, the point $\omega_i(t)$ lies on the frontier of a barrier around some terminal c_k, and ω_i turns *toward* c_k at t.

The second step shows that none of the routes cross over. For if ω_i and ω_j cross over, then they form a polygon as shown in Figure 5-1(v). The portion of ω_i in this polygon lies on the opposite side of ω_j from a_i and b_i. Similarly, the portion of ω_j in this polygon lies on the opposite side of ω_i from a_j and b_j. Like all polygons, this one has three or more internal angles of measure less than π. Two of these can lie at the points where ω_i and ω_j intersect, but at the third angle x one of the routes, say ω_i, turns toward the inside of the polygon. Let c_k be the terminal whose barrier ω_i bends around at x. Then the line segment from x to c_k lies within this barrier, and its length is $|i - k|$. This line segment intersects ω_j. Moreover, the terminals of ω_j lie on the opposite side of ω_i from c_k. Therefore $|j - k|$ is greater than $|i - k|$, and so ω_j comes closer than $|j - k|$ units to c_k. But this means ω_j enters the barrier presented to it by c_k.

The third step shows that the routes ω_i stay at least one unit apart. Suppose ω_i comes within $d < 1$ units of ω_j, as shown in Figure 5-1(vi). Then either a terminal of one comes within d units of the other, contradicting evasiveness, or else there is a point x at which one route, say ω_i, lies within d units of the other, but turns away from it. Let c_k be the terminal whose barrier ω_i bends around at x. Then ω_i turns toward c_k at x, and since ω_j does not cross over ω_i, it follows that c_k lies on the opposite side of ω_i from the terminals of ω_j. Hence the barrier for ω_j around c_k has radius at least $|i - k| + 1$. We have $\|x - c_k\| = |i - k|$, and hence by the triangle inequality for norms, ω_j comes within $|i - k| + d$ units of c_k. Again ω_j enters the barrier presented to it by c_k. We conclude that the routes ω_j actually do stay one unit apart.

Outline of the construction

Many ideas from river routing in a channel carry over to the design routing problem. The first insight that applies is this: a feasible wire must stay far enough away from the fringes, other than its terminals, to allow the other wires to be routed.

We express this constraint in our model by saying that a route of a wire ω is **evasive** in a design $\Omega \ni \omega$ if every nontrivial straight half-cut for that route is safe in Ω. Evasiveness alone does not guarantee feasibility, however. An evasive link can be divisive, and it can have self-intersections. Moreover, it might not leave enough space for wires to pass between two different portions of itself. But if one chooses a canonical, evasive route of minimum length, these problems go away; no two parts of the route are close together except where it is necessary. A route α of a wire ω in a safe design Ω is called **ideal** for Ω if α is canonical, evasive in Ω, and has minimum length among all routes of ω that are evasive in Ω.

To prove that the ideal routes form a design, we analyze the points at which they turn. Intuitively, wherever an ideal link has a joint, it is being pushed away from a fringe by the evasiveness condition. As we show in Section 5B, there is a marginally safe half-cut from that fringe to the joint, and it lies interior to the angle made by the link at that joint. (Ideal links are piecewise straight.) From this result we can deduce many properties of the ideal routes. For example, if the ideal routes of two wires were to cross over, they would form a loop, and one of the two links would turn toward the inside of the loop. But then the marginally safe half-cut would contain an *unsafe* nontrivial half-cut for the other link, contradicting the evasiveness of that link. Therefore the ideal routes do not cross. A similar argument shows that they have no self-intersections, and are actually wires.

Furthermore, *ideal wires are sufficiently separated.* For suppose that two ideal wires were to come closer than the mean of their widths, causing their extents to overlap. Then either their terminals would be too close, or else one wire would turn away from the other at a point where they were close. Concatenating the marginally safe half-cut to that turning point with a short straight path to the other wire yields an unsafe, nondegenerate, bent half-cut for the second ideal wire, and thus by Lemma 4f.6, an unsafe, nondegenerate (and hence nontrivial), straight half-cut. Again, this would contradict the evasiveness of the second wire. If one carefully applies this argument to different parts of the same wire, one can also show that ideal wires are self-avoiding. These are the main steps in the proof that the ideal routes form a proper design.

5A. Construction of Ideal Routes

The result of this section is quite simple: every wire in a safe sketch has an ideal route. We prove this result in two steps. Given a wire in a safe sketch, we first attempt to construct an evasive route for it. If this construction were to fail, we show, the sketch could not be safe. Then we prove that the family of evasive routes for the wire has a minimum-length, canonical element: an ideal route.

In my initial attempts to prove a routability theorem, I tried to construct evasive routes of wires by pushing the wires away from the fringes from which they had unsafe straight half-cuts. I hoped to show that if the process failed to converge, then the wire had straight, unsafe, nontrivial half-cuts pushing it from both sides, and these half-cuts combined to form an unsafe, bent, major cut. By a result like Corollary 4f.5, the existence of this cut would contradict the safety of the design. These attempts failed, partly due to the difficulty of defining the "sides" of a wire in the plane. But in a blanket, where the sides of a simple link are well defined, the idea works. For every wire in a safe sketch we identify left and right *forbidden zones* in the blanket, and show that they do not intersect. Further maneuvering shows that the wire has an evasive route, which in turn yields an ideal route.

Forbidden zones

The blanket provides us with a spatial characterization of evasiveness. Given a wire ω and a lifting $\tilde{\omega}$ of that wire, we identify two *forbidden zones*, one to the left of $\tilde{\omega}$ and one to the right. A route of ω need only have a lifting in $[\tilde{\omega}]_L$ that avoids these zones in order to be evasive. It follows that ω has an evasive route if the forbidden zones for $\tilde{\omega}$ do not separate the terminals of $\tilde{\omega}$.

Definition 5a.1. Let Ω be a design on a sheet S, and let $p\colon M \to S$ be the covering map. Let $\tilde{\omega}$ be a lift of a wire $\omega \in \Omega$. A straight half-link $\tilde{\sigma}$ in M is **forbidden** to $\tilde{\omega}$ if for some link $\tilde{v} \in [\tilde{\omega}]_L$ the path $p \circ \tilde{\sigma}$ is an unsafe, nontrivial half-cut for $p \circ \tilde{v}$. The **left-hand (right-hand) forbidden zone** for $\tilde{\omega}$ is the set of all endpoints $\tilde{\sigma}(1)$ of the forbidden half-links $\tilde{\sigma}$ for $\tilde{\omega}$ with $\tilde{\sigma}(0) \in \mathit{left}(\tilde{\omega})$ (or $\tilde{\sigma}(0) \in \mathit{right}(\tilde{\omega})$).

The forbidden zones for $\tilde{\omega}$ depend only on its link-homotopy class, or equivalently, its terminals. The choice of the link $\tilde{v}$ is also irrelevant, provided that it passes through $\tilde{\sigma}(1)$. This point will be clarified in Lemma 5a.2.

Requiring that $p \circ \tilde{\sigma}$ be nontrivial is equivalent to requiring that $\tilde{\sigma}(0)$ not lie on a terminal of $\tilde{\omega}$. Therefore every forbidden half-link $\tilde{\sigma}$ for $\tilde{\omega}$ satisfies either $\tilde{\sigma}(0) \in \mathit{left}(\tilde{\omega})$ or $\tilde{\sigma}(0) \in \mathit{right}(\tilde{\omega})$, and hence contributes to one of the forbidden zones for $\tilde{\omega}$. Conversely, every sufficiently short straight half-link $\tilde{\sigma}$ is forbidden unless it shares a terminal with $\tilde{\omega}$. For if $\|\tilde{\sigma}\| < \mathit{width}(\omega)/2$, then then $\mathit{cap}(p \circ \tilde{\sigma}) < 0$, and consequently $p \circ \tilde{\sigma}$ is unsafe. Therefore every point sufficiently close to a fringe of M that is not a terminal of $\tilde{\omega}$ belongs to a forbidden zone of $\tilde{\omega}$.

Connection with evasiveness

The first consequence of Definition 5a.1 is that one can find an evasive route of ω by finding a link that is homotopic to $\tilde{\omega}$ and avoids its forbidden zones.

Lemma 5a.2. *Let ω be a wire with lift $\widetilde{\omega}$. The projection of a link $\widetilde{\rho} \in [\widetilde{\omega}]_L$ is evasive if and only if $\widetilde{\rho}$ avoids the forbidden zones of $\widetilde{\omega}$.*

Proof. Let p denote the covering map. First we tackle the "if" direction. Suppose that $p \circ \widetilde{\rho}$ is not evasive; let σ be an unsafe, straight, nontrivial half-cut for $\widetilde{\rho}$ at t. Let $\widetilde{\sigma}$ be a lift of σ with $\widetilde{\sigma}(1) = \widetilde{\rho}(t)$. Then $\widetilde{\sigma}$ is a forbidden half-link for $\widetilde{\omega}$ (take $\widetilde{v} = \widetilde{\rho}$), and hence $\widetilde{\sigma}(1) = \widetilde{\rho}(t)$ lies in a forbidden zone of $\widetilde{\omega}$.

Now we prove the "only if" direction. If $\widetilde{\rho}$ enters a forbidden zone of $\widetilde{\omega}$, then we have $\widetilde{\sigma}(1) = \widetilde{\rho}(r)$ for some forbidden half-link $\widetilde{\sigma}$ and some point $r \in (0, 1)$. Hence for some link $\widetilde{v} \in [\widetilde{\omega}]_L$ and some point $t \in I$, the path $p \circ \widetilde{\sigma}$ is an unsafe, straight, nontrivial half-cut for $p \circ \widetilde{v}$ at t. But the half-cuts $p \circ \widetilde{\sigma}$ for $p \circ \widetilde{\rho}$ at r, and $p \circ \widetilde{\sigma}$ for $p \circ \widetilde{v}$ at t, are akin (Definition 4d.1); they have the same flow, and neither is trivial. And obviously they have the same capacity. Therefore $p \circ \widetilde{\sigma}$ is an unsafe, straight, nontrivial half-cut for $p \circ \widetilde{\rho}$ at r, and $p \circ \widetilde{\rho}$ is not evasive. □

Next we show that the forbidden zones for a link do not intersect. The construction is illustrated in Figure 5a-1.

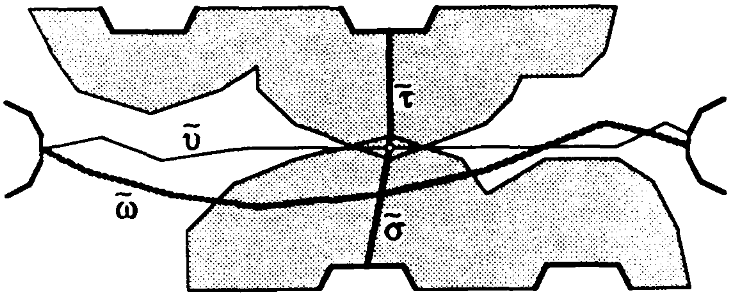


Figure 5a-1. *In a safe design, forbidden zones are disjoint.* Here $\widetilde{\omega}$ lifts a wire ω. If its forbidden zones intersect, some point z (small circle) is touched by forbidden half-links $\widetilde{\sigma}$ and $\widetilde{\tau}$ from both sides. The concatenation $\widetilde{\chi}$ of these half-links cuts $\widetilde{\omega}$. Hence its projection is a bent cut χ that makes a necessary crossing of ω, and the half-links corresponding to this crossing are unsafe. We infer that the bent cut itself is unsafe and major.

Lemma 5a.3. *Let ω be a wire in a safe design, and let $\widetilde{\omega}$ lift ω. The forbidden zones for $\widetilde{\omega}$ are disjoint.*

Proof. Call the design Ω. Let S be the sheet of Ω, let M be its blanket, and let $p\colon M \to S$ be the covering map. Suppose $\widetilde{\omega}$ splits M into scraps L and R, and let z be a point in both forbidden zones of $\widetilde{\omega}$. Let $\widetilde{v}$ be any link in $[\omega]_L$ that passes through z; say $\widetilde{v}(t) = z$. Applying Definition 5a.1 to z, we find straight half-links $\widetilde{\sigma}$ and $\widetilde{\tau}$ ending at z such that $\widetilde{\sigma}(0) \in L$, $\widetilde{\tau}(0) \in R$, and both $\sigma = p \circ \widetilde{\sigma}$ and $\tau = p \circ \widetilde{\tau}$ are unsafe, straight, nontrivial half-cuts for $p \circ \widetilde{v}$ at t. Thus if A and B are the terminals of σ and τ respectively, we have

$$\begin{aligned} \mathit{flow}(\sigma,\Omega) > \mathit{cap}(\sigma,\Omega) &= \|\sigma\| - \mathit{width}(A)/2 - \mathit{width}(\omega)/2; \\ \mathit{flow}(\tau,\Omega) > \mathit{cap}(\tau,\Omega) &= \|\tau\| - \mathit{width}(B)/2 - \mathit{width}(\omega)/2. \end{aligned}$$

We find an unsafe, major, simple cut in Ω, which by Corollary 4f.5 implies that Ω is not safe. Let $\widetilde{\chi}$ equal $\widetilde{\sigma} \star \widetilde{\tau}_{1:0}$, and put $\chi = p \circ \widetilde{\chi}$. Then χ is a bent cut. Because σ and τ are nontrivial, their terminals lie wholly in opposite scraps of $\widetilde{\omega}$, and hence the link $\widetilde{\chi}$ cuts $\widetilde{\omega}$. Therefore the crossing $(\frac{1}{2}, t)$ of χ by $p \circ \widetilde{v}$ is necessary. By Proposition 4d.2, we have

$$\begin{aligned} \mathit{flow}(\chi,\Omega) &\geq \mathit{flow}(\chi_{0:1/2},\Omega) + \mathit{flow}(\chi_{1:1/2},\Omega) + \mathit{width}(\omega) \\ &= \mathit{flow}(\sigma,\Omega) + \mathit{flow}(\tau,\Omega) + \mathit{width}(\omega) \\ &> \|\sigma\| + \|\tau\| - \mathit{width}(A)/2 - \mathit{width}(B)/2. \end{aligned}$$

Because $\|\chi\| = \|\sigma\| + \|\tau\|$, the final quantity is just the capacity of χ. We conclude that $\mathit{flow}(\chi,\Omega)$ exceeds $\mathit{cap}(\chi,\Omega)$, making χ unsafe. Because its flow is nonzero, χ is nonempty in Ω. And since χ is simple, Lemma 4e.3 implies that χ is nondegenerate. Hence χ is major in Ω. □

Forbidden zones are made of barriers

The fact the forbidden zones for a link $\widetilde{\omega}$ do not intersect does not immediately imply that the zones can be avoided by a piecewise linear link between the terminals of $\widetilde{\omega}$. To construct this link, we analyze the forbidden zones themselves.

First we chop the fringes of the blanket into small pieces. Let Ω be a design on the sheet S, and let M be a blanket of S with covering map $p\colon M \to S$. Let ω be a wire in Ω, and let $\widetilde{\omega}$ be a lift of ω to M. Choose ϵ smaller than the minimum dimension of the fringes of S, and cover *Bd* S with connected open sets of size ϵ or less. Because *Bd* S is compact, finitely many sets suffice. Each set in the resulting open cover is contractible and locally path-connected, and hence can be lifted to M by Proposition 2b.8.

Each lift of these fringe pieces gives rise to some part of a forbidden zone. For each lift U of a set V in the open cover, we define the **barrier** for $\widetilde{\omega}$ **growing from** U to be the set of endpoints $\sigma(1)$ of forbidden half-links $\widetilde{\sigma}$ for $\widetilde{\omega}$ with $\widetilde{\sigma}(0) \in U$. The **base** of this barrier is the fringe containing U. A barrier for $\widetilde{\omega}$ is a **left-hand barrier** or **right-hand barrier** according to whether its base lies in $\mathit{left}(\widetilde{\omega})$ or $\mathit{right}(\widetilde{\omega})$.

Both forbidden zones for $\widetilde{\omega}$ are unions of barriers for $\widetilde{\omega}$. The following lemma relates the barriers and zones more directly.

Lemma 5a.4. *Let $\widetilde{\omega}$ lift a wire ω in the design Ω; let Z be the right-hand forbidden zone of $\widetilde{\omega}$ and put $R = \mathit{right}(\widetilde{\omega})$. Then*

(1) *every barrier for $\widetilde{\omega}$ is a lift of the inside of a polygon;*
(2) *only finitely many right-hand barriers for $\widetilde{\omega}$ intersect $\widetilde{\omega}$; and*
(3) *the union X of those barriers satisfies $Z - R \subseteq X \subseteq Z$.*

Proof. Say ω is a wire in the design Ω on the sheet S. Call a point x in a flat manifold **visible** from a set U in that manifold if there is a straight path from x to a point of U. Combining Definition 5a.1 with the definition of the flow across a half-cut, we characterize barriers as follows:

Claim: *A point $x \in M$ is in the barrier for $\widetilde{\omega}$ growing from U if and only if there is a straight half-link $\widetilde{\sigma}$ from U to x, and a half-link $\widetilde{\alpha}$ from the fringe containing $\widetilde{\omega}(1)$ to x, such that*

$$\begin{aligned} \mathit{flow}(p \circ (\widetilde{\sigma} \star \widetilde{\alpha}_{1:0}), \Omega) &> \mathit{cap}(p \circ \widetilde{\sigma}) \\ &= \|p \circ \widetilde{\sigma}\| - \mathit{width}(F)/2 - \mathit{width}(\omega)/2, \end{aligned}$$

where F is the fringe of S containing $p(U)$.

The quantity $f = \mathit{flow}(p \circ (\widetilde{\sigma} \star \widetilde{\alpha}_{1:0}), \Omega)$ depends only on the fringe containing $\widetilde{\sigma}(0)$, because if this and $\widetilde{\omega}$ are held fixed, all the links $p \circ (\widetilde{\sigma} \star \widetilde{\alpha}_{1:0})$ are link-homotopic. Hence the barrier for $\widetilde{\omega}$ growing from U is the set of points in $M - \mathit{Bd}\, M$ visible from U whose distance from U (in the wiring norm) is less than $f + \mathit{width}(F)/2 + \mathit{width}(\omega)/2$, proving the claim.

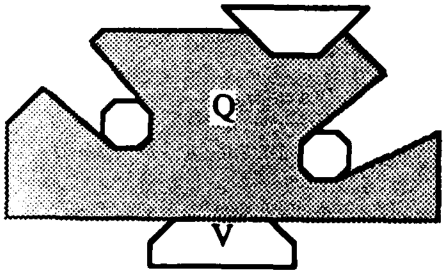


Figure 5a-2. *The shape of a barrier.* We break up the forbidden zones into barriers, which are lifts of polygonal regions like the shaded set Q shown here. This region is the set of points interior to the sheet that are visible from the striped set V, and lie within a certain distance of it.

Now put $V = p(U)$, and let Q denote the set of points in $S - \mathit{Bd}\, S$ visible from V whose distance from $p(U)$ is less than $f + \mathit{width}(F)/2 + \mathit{width}(\omega)/2$. Because the wiring norm is piecewise linear, and the fringes are polygons, Q is bounded by line segments. Because V is connected and open in $\mathit{Bd}\, S$, the set Q is connected and open; because V is smaller than any fringe of S, the closure of Q has no "holes". Hence Q is the inside of a polygon in S, and $\mathit{Cl}\, Q$ is simply connected. By Lemma 2b.8, $p^{-1}(\mathit{Cl}\, Q)$ consists of disjoint copies of $\mathit{Cl}\, Q$. One of these, call it P', contains U. Put $P = P' \cap p^{-1}(Q)$. One easily checks that P is the barrier for $\widetilde{\omega}$ growing from U, proving (1).

We now prove part (2) by showing that $\widetilde{\omega}$ intersects only finitely many of its barriers. Given any x in M, choose a polygonal neighborhood O of $p(x)$ in S, and define the neighborhood N_x of x to be the component of $p^{-1}(O)$ that contains x. For each set Q described above, the intersection $O \cap Q$ has finitely many components, and hence $N_x \cap p^{-1}(Q)$ has finitely many components. Therefore N_x intersects only finitely many components of $p^{-1}(Q)$. Every barrier for $\widetilde{\omega}$ is a component of $p^{-1}(Q)$ for some set Q described above, and the number of such sets Q is finite. Hence N_x intersects only finitely many barriers of $\widetilde{\omega}$. Because $Im\, \widetilde{\omega}$ is compact, it can be be covered by finitely many neighborhoods N_x, and hence intersects only finitely many barriers, which proves (2). Let $\{P_i\}$ be the set of right-hand barriers of $\widetilde{\omega}$ that intersect $\widetilde{\omega}$.

To show (3) we prove the inclusions $Z - R \subseteq \bigcup_i P_i \subseteq Z$. Certainly every barrier P_i is a subset of Z. On the other hand, if x is a point in $Z - R$, then x is in some right-hand barrier P for $\widetilde{\omega}$; we have $x = \widetilde{\sigma}(1)$ where $\widetilde{\sigma}(0) \in R$ and $\widetilde{\sigma}$ is a forbidden half-link for $\widetilde{\omega}$. This half-link must intersect $\widetilde{\omega}$; say $\widetilde{\sigma}(s) \in Im\, \widetilde{\omega}$ where $s > 0$. Then $\widetilde{\sigma}_{0:s}$ is a forbidden half-link for $\widetilde{\omega}$, so $\alpha(s) \in P$. Therefore $P = P_i$ for some i because P intersects $\widetilde{\omega}$. This proves part (3). □

So every barrier in M is a connected component of the inverse image (under the covering map) of an open set in S. Hence barriers are open. It follows that forbidden zones, which are unions of barriers, are themselves open.

Detours

To construct a link that avoids forbidden zones, we start with any link and repeatedly insert detours around barriers. Lemma 5a.4 shows that the number of barriers we need to consider is finite. This proof technique is not the most elegant, but it works. The following definition aids in describing the process of inserting detours.

Definition 5a.5. Let α be a simple link in a blanket M, and let P be an open subset of M. A **detour of α around P** is a simple link $\alpha' \in [\alpha]_L$ such that

$$Im\, \alpha' \subseteq (Im\, \alpha \cup Fr\, P) - P.$$

The detour α' is **leftward** if $P \subseteq right(\alpha')$ and α' does not intersect $right(\alpha)$.

The next lemma takes care of the induction step by showing that two detours can be combined into one. Its use of the Detour Lemma gives that result its name.

Lemma 5a.6. *Let β and β' be leftward detours of a simple link α around the open sets P and P'. Then there is a leftward detour of α around $P \cup P'$.*

Proof. We apply the Detour Lemma (3c.3) to the links β and β', obtaining a simple link γ that is link-homotopic to both β and β', and hence to α. Part of that lemma

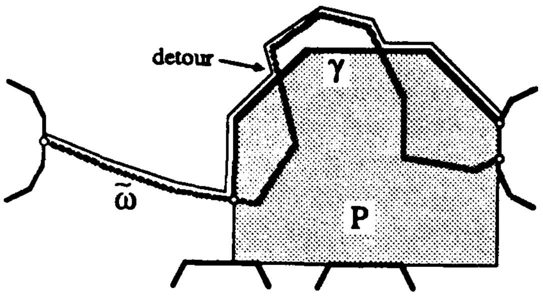


Figure 5a-3. *Making a detour around a single barrier.* If the link $\widetilde{\omega}$ intersects the barrier P, we splice a section of $Fr\,P$ into ω to form a simple path β, and then trim β to form a simple link γ. Applying the Detour Lemma to γ and $\widetilde{\omega}$, we get a detour of $\widetilde{\omega}$ around P.

states that the right-hand scrap of γ contains those of β and β', so both P and P' lie to the right of γ, as does the right-hand scrap of α. Hence if γ is a detour of α around $P \cup P'$, it is leftward. The other claim of Lemma 3c.3 is that $Im\,\gamma$ lies within $Im\,\beta \cup Im\,\beta'$. Therefore

$$\begin{aligned} Im\,\gamma &\subseteq Im\,\beta \cup Im\,\beta' \\ &\subseteq (Im\,\alpha \cup Fr\,P) \cup (Im\,\alpha \cup Fr\,P') \\ &= Im\,\alpha \cup (Fr\,P \cup Fr\,P'). \end{aligned}$$

An elementary topological calculation shows that $Fr(P \cup P') = (Fr\,P \cup Fr\,P') - (P \cup P')$. Since $Im\,\gamma$ does not intersect P or P', it follows that

$$Im\,\gamma \subseteq \big(Im\,\alpha \cup Fr(P \cup P')\big) - (P \cup P').$$

Therefore γ is a detour of α around $P \cup P'$. □

Now we consider the basis case: making a single detour around a barrier. Because barriers are polygons, this is manageable, but somewhat tedious.

Lemma 5a.7. *Let $\widetilde{\omega}$ be a lift of a wire in a safe design Ω. Then for every right-hand barrier P of $\widetilde{\omega}$, there is a leftward detour of $\widetilde{\omega}$ around P.*

Proof. Let L and R denote $left(\widetilde{\omega})$ and $right(\widetilde{\omega})$; let Z be the right-hand forbidden zone of $\widetilde{\omega}$. The set P is homeomorphic to the inside of a simple polygon by Lemma 5a.4, and we may assume it intersects $Im\,\widetilde{\omega}$. To construct the detour, we replace the parts of $\widetilde{\omega}$ that pass through P by a path along $Fr\,P$. Put $s = \inf \widetilde{\omega}^{-1}(Cl\,P)$ and $t = \sup \widetilde{\omega}^{-1}(P)$. Now let τ be a simple sublink within P from $\widetilde{\omega}(s)$ to $\widetilde{\omega}(t)$, and consider the link

$$\alpha = \widetilde{\omega}_{0:s} \star \tau \star \widetilde{\omega}_{t:1}.$$

It separates the polygon $Fr\,P$ into two components; let τ' be the path in $Fr\,P$ from $\widetilde{\omega}(s)$ to $\widetilde{\omega}(t)$ that lies to the left of α, and hence keeps P to the right. Define a path β by

$$\beta = \widetilde{\omega}_{0:s} \star \tau' \star \widetilde{\omega}_{t:1}.$$

The path β is piecewise linear, and lies in $Im\,\widetilde{\omega} \cup Fr\,P$. We show that β intersects no fringe F other than the terminals of $\widetilde{\omega}$. If F lies completely in L, then all points close enough to F are in the left-hand forbidden zone of $\widetilde{\omega}$, which P cannot touch by Lemma 5a.3. Furthermore, β cannot intersect any fringe that lies completely in R, because β lies to the left of α, while the fringes of R lie to the right of α. (The link α is link-homotopic to $\widetilde{\omega}$, and hence by Proposition 3c.4, separates the fringes of M as $\widetilde{\omega}$ does.) Therefore β runs between the terminals of $\widetilde{\omega}$. Its middle may intersect these terminals, however.

We now convert β into a simple link. Say that β runs from fringe X to fringe Y. Set $s = \sup \beta^{-1}(X)$ and $t = \inf\big((s,1] \cap \beta^{-1}(Y)\big)$. Then $\gamma = \beta_{s:t}$ is a simple link between X and Y, and is piecewise linear; its image also lies in $Im\,\widetilde{\omega} \cup Fr\,P$. Then γ is a detour of $\widetilde{\omega}$ around P: it is link-homotopic to $\widetilde{\omega}$ because it runs between the same terminals, and its image lies in $(Im\,\widetilde{\omega} \cup Fr\,P) - P$. By the construction of β, there are some points of P that lie in *right*(γ). Hence P, being connected, must lie entirely in *right*(γ).

Still, γ may not be a leftward detour, because it may enter the right-hand scrap of $\widetilde{\omega}$. But the Detour Lemma (3c.3) applied to $\widetilde{\omega}$ and γ solves this problem. □

All that remains is to put the pieces together.

Proposition 5a.8. *Every wire in a safe design has an evasive route.*

Proof. Let Ω be a safe design on a sheet S, and let ω be a wire in Ω. Let M be a blanket of S with covering map $p\colon M \to S$. Lift ω to a simple link $\widetilde{\omega}$ in M. We construct an evasive route δ by finding detours of $\widetilde{\omega}$ around its forbidden zones.

Let L and R be the left and right scraps of $\widetilde{\omega}$, and let Z be the right-hand forbidden zone of $\widetilde{\omega}$. Apply Lemma 5a.4 to find barriers P_i such that $Z - R \subseteq \bigcup P_i$. For each i, apply Lemma 5a.7 to obtain a leftward detour of $\widetilde{\omega}$ around P_i. Repeated application of Lemma 5a.6 then gives us a leftward detour of $\widetilde{\omega}$ around $\bigcup P_i$. Call it δ. Because δ does not enter R, and avoids $Z - R$, it must avoid Z entirely. Now let Z' be the right-hand forbidden zone of $\widehat{\delta}$, which is the left-hand forbidden zone of $\widetilde{\omega}$. Decompose $Z - L$ into barriers P_i', and apply the same technique to find a leftward detour η of $\widehat{\delta}$ around Z'. By construction, η avoids Z', and $Im\,\eta \subseteq Im\,\widehat{\delta} \cup Fr\,Z'$.

I claim that η avoids Z as well as Z'. Let x be a point of $Im\,\eta$. If $x \in Im\,\widehat{\delta}$, then $x \in Im\,\delta$, and hence $x \notin Z$. So assume that $x \in Fr\,Z'$. If x were in Z, then because Z is open (Lemma 5a.4), Z and Z' would intersect, contradicting Lemma 5a.3. Therefore η is a simple link in $[\widetilde{\omega}]_L$ that avoids the forbidden zones of $\widetilde{\omega}$. By Lemma 5a.2, the route $p \circ \eta$ of ω is evasive in Ω. □

Ideal routes

Building on Proposition 5a.8, we now complete the construction of ideal routes for wires in a safe design. All that remains is to show that among the evasive routes of a wire there is one of minimum length. The Reparameterization Lemma (3d.1) allows us to make this route canonical.

Proposition 5a.9. *Every wire in a safe design has an ideal route.*

Proof. Let Ω be a safe design on a sheet S; let ω be a wire in Ω with lifting $\widetilde{\omega}$. Let M be the blanket for S, and denote by Z the union of the forbidden zones of $\widetilde{\omega}$. Let X and Y denote the fringes of M containing $\widetilde{\omega}(0)$ and $\widetilde{\omega}(1)$, respectively. We consider the family Λ of canonical, evasive routes of ω. Since reparameterizing a path does not affect its evasiveness or its arc length (see Lemma 3d.1), Proposition 5a.8 shows that Λ is nonempty, and $l = \inf\{\, |\lambda| : \lambda \in \Lambda \,\}$. By Proposition 2c.8, the collection Λ contains a uniformly convergent sequence $\langle \alpha_k \rangle_{k=1}^{\infty}$ whose limit α has euclidean arc length at most l.

I claim that α has a lifting $\widetilde{\alpha}\colon X \rightsquigarrow Y$ that avoids Z. Let β be any lift of α to M. By Lemma 3a.7, the paths α_k have lifts β_k that converge uniformly to β. In particular, there is a constant K such that the paths $\{\beta_k\}_{k>K}$ run between the same fringes of M. Let h be a covering transformation that carries those fringes onto the fringes of $\widetilde{\omega}$. (One must exist, since the paths α_k have liftings in $[\widetilde{\omega}]_L$.) Then $h \circ \beta_k$ is a path in $M - Z$ for each $k > K$, and since forbidden zones are open, by Lemma 5a.4, the limit $h \circ \beta$ of the sequence $\langle h \circ \beta_k \rangle$ also avoids Z. Write $h \circ \beta$ as $\widetilde{\alpha}$. The fringes of M are closed, so the endpoints of β lie on the terminals of β_k, and therefore $\widetilde{\alpha}$ has the same terminals as $\widetilde{\omega}$.

Now we convert α into a canonical route of ω. Put $s = \sup \widetilde{\alpha}^{-1}(X)$ and $t = \inf\bigl((s,1] \cap \widetilde{\alpha}^{-1}(Y)\bigr)$. Then $\widetilde{\alpha}_{s:t}$ intersects X and Y at its endpoints alone. It cannot intersect any other fringe of M, for it would have to cross Z to do so. Therefore $\widetilde{\alpha}_{s:t}$ is a link, and $\alpha_{s:t}$ is an evasive route of ω. Using Lemma 3d.1, let $\widetilde{\gamma}$ be a canonical version of $\widetilde{\alpha}_{s:t}$; it has the same image, arc length, and path class. One can check that $\widetilde{\gamma}^{-1}(X) = \{0\}$ and $\widetilde{\gamma}^{-1}(Y) = \{1\}$, which makes $\widetilde{\gamma}$ a link. Hence its projection γ is a canonical, evasive route of ω; in symbols, $\gamma \in \Lambda$. We have $|\gamma| = |\widetilde{\alpha}_{s:t}| \le |\alpha| \le l$, and hence $|\gamma| = l$. Therefore γ has minimum length among all evasive routes of ω. In other words, γ is an ideal route of ω. □

5B. Ideal Routes Are Taut

Now we begin the process of characterizing ideal routes. Top priority is to show that liftings of ideal routes are simple, so that our results about simple links will apply to them. At the same time we prove that an ideal route has vertices only where it bends around its barriers. Then we prove a key technical lemma: every straight half-cut for an ideal route is either trivial or semisimple. That fact enables us to prove that ideal routes are *taut*: wherever one turns, it is supported by a straight, marginal, nondegenerate half-cut. Later sections will use these half-cuts to demonstrate that other ideal routes, which are evasive, cannot approach this one. And as we show in Section 6B, tautness implies that the ideal route cannot be made any shorter without becoming infeasible.

Getting off the ground

One easy result is that lifts of ideal routes are injective. For if β lifts an ideal route α of ω and $\beta(s) = \beta(t)$, where $s \leq t$, then $\alpha_{0:s} \star \alpha_{t:1}$ is an evasive route of ω, and its arc length is $1 - t + s$ times that of α because α is canonical. Since α has minimum length among the evasive routes of ω, and $|\alpha| > 0$, it follows that $s = t$. The first task is to prove that these liftings are piecewise linear, and therefore simple. We start with a technical lemma.

Lemma 5b.1. *Let α be a link in a blanket. There are simple links $\beta, \gamma \in [\alpha]_L$ such that $Im\,\alpha \in left(\beta) \cap right(\gamma)$.*

Proof. For each point $t \in I$, let $\alpha_t \in [\alpha]_L$ be a simple link that passes through $\alpha(t)$. By modifying α_t in the neighborhood of $\alpha(t)$, find links β_t and γ_t in $[\alpha]_L$ such that $\alpha(t) \in left(\beta_t) \cap right(\gamma_t)$. Write $L_t = left(\beta_t)$ and set $U_t = \alpha^{-1}(L_t)$; similarly, put $R_t = right(\gamma_t)$ and $V_t = \alpha^{-1}(R_t)$. Since $t \in U_t \cap V_t$ and the sets U_t and V_t are open, the collection $\{U_t \cap V_t\}$ is an open cover of I. Because I is compact, it has a finite subcover

$$U_{t_1} \cap V_{t_1}, \ldots, U_{t_n} \cap V_{t_n}.$$

Then $Im\,\alpha \subset \bigcup_{i=1}^{n} L_{t_i}$, and also $Im\,\alpha \subset \bigcup_{i=1}^{n} R_{t_i}$. By iterative application of the Detour Lemma (3c.3), there is a simple link $\gamma \in [\alpha]_L$ whose right-hand scrap contains those of $\gamma_{t_1}, \gamma_{t_2}, \ldots, \gamma_{t_n}$. A symmetrical argument yields a simple link $\beta \in [\alpha]_L$ whose left-hand scrap contains those of $\beta_{t_1}, \beta_{t_2}, \ldots, \beta_{t_n}$. It follows that $Im\,\alpha$ lies left of β and right of γ. □

Turning points of ideal routes

The proof that shows ideal routes are piecewise linear, which we are about to begin, also characterizes the points at which they turn. Suppose that α is a simple link in a blanket M, and that α turns at $x \in (0,1)$. Let A and B be the scraps of $M - Im\,\alpha$. One of these scraps, say A, contains points internal to the angle made by α at x. Let C be a subset of M. If C intersects A but not B, then α **turns toward** C at x; if C intersects B but not A, then α **turns away from** C at x.

We prove the very intuitive fact that at each joint of an ideal route, it bends around the vertex of a barrier, and hence turns toward that barrier. The idea behind the proof is that where an ideal route is not constrained by the vertex of a barrier, it is elastic and hence linear. We have worked through a similar proof before: see Lemma 3d.5.

Lemma 5b.2. *Let α be an ideal route with lift $\tilde{\alpha}$. The link $\tilde{\alpha}$ is simple, and if α is not straight at $x \in (0,1)$, then some barrier P for $\tilde{\alpha}$ has a vertex at $\tilde{\alpha}(x)$, and $\tilde{\alpha}$ turns toward P at x.*

Proof. Let Ω be a safe design on a sheet S, and let α be an ideal route of a wire $\omega \in \Omega$. Let M be a blanket of S with covering map $p\colon M \to S$. Let Z be the unicn of the forbidden zones for $\tilde{\alpha}$. Because α is an evasive route of ω, the link $\tilde{\alpha}$ avoids Z. By Lemma 5b.1, there are links β and γ in M such that $Im\,\alpha \in \mathit{left}(\beta) \cap \mathit{right}(\gamma)$.

Suppose that $\tilde{\alpha}$ is not straight at the point $x \in (0,1)$. Let U be a neighborhood of $\tilde{\alpha}(x)$ that intersects neither β nor γ. By Lemma 5a.4, the set $U \cap Z$ is the intersection of U with finitely many barriers $P_1, \ldots, P_n$ of β and γ. Also by Lemma 5a.4, these barriers are polygonal. Hence by restricting U we may assume that U contains no vertex of a barrier P_i except those lying at $\tilde{\alpha}(x)$. Then the situation is as shown in Figure 5b-1. We may assume that $U \cap Bd\,M$ is empty and that $p(U)$ is convex. Because $\tilde{\alpha}$ is continuous and evasive, there is an interval (s,t) containing x such that $\tilde{\alpha}[s,t] \subset U - Z$.

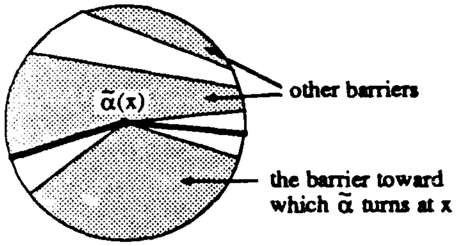


Figure 5b-1. *Where an ideal link turns.* Within a neighborhood small enough to include only one barrier vertex, a lifting $\tilde{\alpha}$ of an ideal route is either straight, or else it turns toward that barrier as shown here.

The straight path $\tilde{\kappa}$ between $\tilde{\alpha}(s)$ and $\tilde{\alpha}(t)$ must intersect Z. For if not, let $\tilde{\alpha}'$ be the result of replacing the subpath $\tilde{\alpha}_{s:t}$ of $\tilde{\alpha}$ with κ. Then $\tilde{\alpha}'$ avoids Z, and hence $\alpha' = p \circ \tilde{\alpha}'$ is evasive. Furthermore, $\alpha' \simeq_L \alpha$ because their lifts are link-homotopic. If $\alpha' \neq \alpha$, then α' is shorter, because a linear path is the only shortest canonical path between two points. (Compare Lemma 3d.3.) But α has the minimum length among all evasive routes of ω. Therefore $\alpha = \alpha'$, so α is straight at x, contrary to assumption. Thus $\tilde{\kappa}$ must intersect some barrier P_i, and this barrier must have a vertex at x. Hence $\tilde{\alpha}$ is straight everywhere except the vertices of the barriers P_i. Since these vertices are finite in number, and $\tilde{\alpha}$ has finite arc length, we conclude that $\tilde{\alpha}$ is piecewise linear. In fact, because $\tilde{\alpha}$ is nonconstant and canonical, it is piecewise straight.

It remains to show that $\tilde{\alpha}$ turns toward P_i at x. Let $\tilde{\sigma}$ be the linear path from $\tilde{\alpha}(s)$ to $\tilde{\alpha}(x)$, and let $\tilde{\tau}$ be the linear path from $\tilde{\alpha}(x)$ to $\tilde{\alpha}(t)$. Then $\tilde{\sigma}$ and $\tilde{\tau}$ do not intersect Z, so by the argument above, we must have $\tilde{\alpha}_{s:x} = \tilde{\sigma}$ and $\tilde{\alpha}_{x:t} = \tilde{\tau}$. We conclude that $\tilde{\alpha}$ turns at x. The barrier P_i has points internal to the angle of $\tilde{\alpha}$

at x, and does not intersect $\widetilde{\alpha}$ because α is evasive. Therefore $\widetilde{\alpha}$ turns toward P_i at x. □

We can extend the notion of turning to the endpoints of a link. Let α be a simple link in a sheet S. We say that α **turns** at $e \in \{0,1\}$ if a straight subpath $\alpha_{e:t}$ of α makes an acute angle with an edge of the fringe containing $\alpha(e)$. A link in a blanket turns wherever its projection does. Let α be a simple link in a blanket M, and let $e \in \{0,1\}$ be a point at which its projection turns. Then we say α turns at e. Let A and B denote the scraps of $M - Im\,\alpha$, and suppose A contains points internal to an acute angle made by α at e. We say that α **turns toward** a set $C \subset M$ at e if C intersects A but not B. Then we obtain the following extension of Lemma 5b.2.

Lemma 5b.3. *Let α be an ideal route with lift $\widetilde{\alpha}$. If α turns at $e \in \{0,1\}$, then some barrier P for $\widetilde{\alpha}$ has a vertex at $\widetilde{\alpha}(e)$, and $\widetilde{\alpha}$ turns toward P at e.*

Proof. Assume without loss of generality that $e = 0$. Let ν be a straight path in the terminal containing $\alpha(0)$ such that $\nu \star \alpha$ has an acute angle at $\alpha(0)$. Let E be the fringe containing $\widetilde{\alpha}(0)$, and let $\widetilde{\nu}$ be a lift of ν satisfying $\widetilde{\nu}(1) = \widetilde{\alpha}(0)$. Then $\widetilde{\alpha}$ turns toward $\widetilde{\nu}(0)$ at 0. Let A be the scrap of $M - Im\,\widetilde{\alpha}$ that contains $\widetilde{\nu}(0)$, and let $Z \subseteq A$ be the forbidden zone for $\widetilde{\alpha}$ on the same side as $\widetilde{\nu}(0)$.

Some barrier $P \subseteq Z$ for $\widetilde{\alpha}$ has a vertex at $\widetilde{\alpha}(0)$. For if not, then because no edge of a barrier in Z can contain $\widetilde{\alpha}(0)$, there is a neighborhood U of $\widetilde{\alpha}(0)$ that does not intersect Z. Let $\widetilde{\sigma}$ be a straight path in this neighborhood from E to $\alpha(s)$ that intersects E perpendicularly. Then $\widetilde{\sigma}$ is shorter than $\widetilde{\alpha}_{0:s}$, and $\widetilde{\sigma} \star \widetilde{\alpha}_{s:1}$ avoids the forbidden zones of $\widetilde{\alpha}$. Projecting to the sheet, one thus obtains an evasive route of α that is shorter than α, contradicting the assumption that α is ideal. □

Straight half-cuts for ideal routes

Now that we know ideal routes are piecewise straight, we can begin to apply our tools to them. One major property of a route that is ideal for a design Ω is that its nontrivial straight half-cuts respect Ω. Actually, they have an even stronger property:. they are semisimple in Ω, which by Proposition 4e.6 implies that they are nondegenerate in Ω and respect Ω strongly.

Proposition 5b.4. *If ω is an ideal route, then every straight half-cut for ρ is either trivial or semisimple.*

Outline of proof. Let σ be a nontrivial straight half-cut for ω at s. Let S be the relevant sheet, and let γ be a straight cut of S such that $\sigma = \gamma_{0:a}$ for some $a \in (0,1)$. Lift ω to a simple link $\widetilde{\omega}$, and lift γ to a straight link $\widetilde{\gamma}$ such that $\widetilde{\gamma}(a) = \widetilde{\omega}(s)$. Let (b,t) be the crossing of $\widetilde{\gamma}$ by $\widetilde{\omega}$ that minimizes b. The half-cut $\gamma_{0:b}$ for ω at t is akin to the half-cut $\gamma_{0:a}$ for ω at s, which is σ. Hence it suffices to prove the lemma in the case $(b,t) = (a,s)$. We find a simple cut χ and a necessary crossing (c,s) of χ by ω such that $\sigma = \chi_{0:c}$. There are five cases to consider, of which four are easy and the fifth requires a further case analysis.

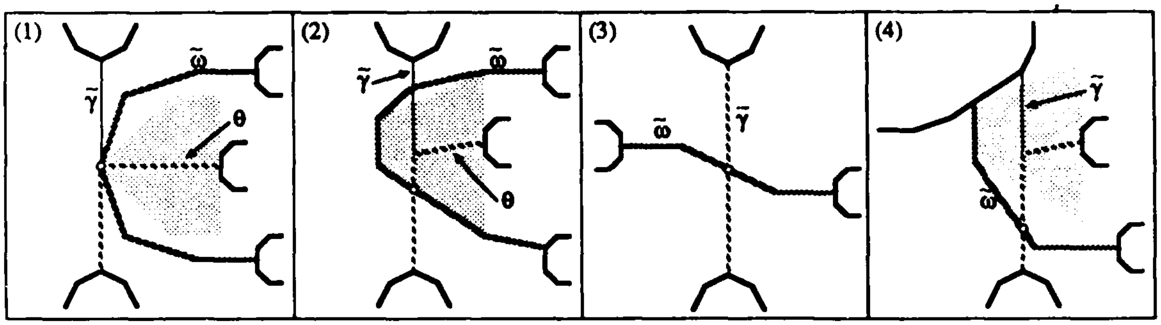


Figure 5b-2. ***Straight half-cuts for an ideal route.*** **Given a nontrivial straight half-cut σ for an ideal route ω, we extend it to a straight cut γ, and lift both the cut and the route to the blanket. The liftings are denoted $\tilde{\gamma}$ and $\tilde{\omega}$. In each of four cases, here labeled (1) through (4), we construct a link (dashed) starting at $\tilde{\gamma}(0)$ that cuts $\tilde{\omega}$. The projection of this bent link is a simple cut that makes σ semisimple. One case is missing here; it is like case (4) but $\tilde{\omega}$ does not turn toward $\tilde{\gamma}(1)$. For this case, see Figure 5b-3.**

(1) The link $\tilde{\omega}$ does not cross over $\tilde{\gamma}$ at s. Then $\tilde{\omega}$ turns away from $\tilde{\gamma}(0)$ at s. Let P be a barrier for $\tilde{\omega}$ having a vertex at $\tilde{\omega}(s)$, such that $\tilde{\omega}$ turns toward P at s. Choose a bent half-link θ in $Cl\,P$ from the base of P to $\tilde{\omega}(s)$. Then the link $\alpha = \tilde{\gamma}_{0:a} \star \theta_{1:0}$ is simple, and its endpoints lie on opposite sides of $\tilde{\omega}$. Since σ and the projection of θ are nontrivial half-cuts, neither $\tilde{\gamma}(0)$ nor $\theta(0)$ lies on a terminal of $\tilde{\omega}$. Hence α actually cuts $\tilde{\omega}$. So ω necessarily crosses $p \circ \alpha$ at s. We set $\chi = p \circ \alpha$ and $c = \frac{1}{2}$.

(2) The link $\tilde{\omega}$ crosses over $\tilde{\gamma}$ at s, and crosses back at some point $\tilde{\gamma}(b) = \tilde{\omega}(t)$, where $b > a$. Assume (b,t) is chosen to minimize $b - a$. Since the path $\tilde{\gamma}_{a:b}$ is shorter than $\tilde{\omega}_{s:t}$, it must intersect a forbidden zone of $\tilde{\omega}$. Choose a half-link θ in this forbidden zone such that $\theta(1) = \tilde{\gamma}(e)$ for some $e \in (a,b)$. Then $\tilde{\gamma}(0)$ and $\theta(1)$ lie on opposite sides of $\tilde{\omega}$, and neither lies on a terminal of $\tilde{\omega}$. Hence $\alpha = \tilde{\gamma}_{0:e} \star \theta_{1:0}$ cuts $\tilde{\omega}$, and so ω makes a necessary crossing with α at $\tilde{\omega}(s)$. Because α is bent and its segments are not parallel, $p \circ \alpha$ is simple. We put $\chi = p \circ \alpha$ and define c by $\alpha(c) = \tilde{\omega}(s)$.

It remains to consider situations in which $\tilde{\omega}$ crosses over $\tilde{\gamma}$ at s and does not cross back. Then $\tilde{\gamma}(0)$ and $\tilde{\gamma}(1)$ lie on opposite sides of $\tilde{\omega}$, and $\tilde{\gamma}$ either cuts or shares a terminal with $\tilde{\omega}$. Since σ is not trivial, the shared terminal cannot contain $\tilde{\gamma}(0)$. Thus we have the following cases.

(3) The link $\tilde{\omega}$ cuts $\tilde{\omega}$. Then the crossing (a, s) of γ by ω is necessary, and we simply put $\chi = \gamma$ and $c = a$.

(4) The link $\tilde{\omega}$ crosses over $\tilde{\gamma}$ at s, and $cross(\tilde{\gamma}, \tilde{\omega}) = 1$; for some $i \in \{0, 1\}$ the point $\tilde{\gamma}(1)$ lies on the fringe containing $\tilde{\omega}(i)$; and $\tilde{\omega}$ turns toward $\tilde{\gamma}(1)$ at a point in (i, s). Then a forbidden zone of $\tilde{\omega}$ intersects the inside of $Im(\tilde{\omega}_{i:s} \star \tilde{\gamma}_{1:a})$, which is a web of one thread. Because there are no fringes in this area, and $\tilde{\omega}$ avoids its forbidden zones, this zone must intersect $\tilde{\gamma}_{1:a}$. We construct χ as in case (2).

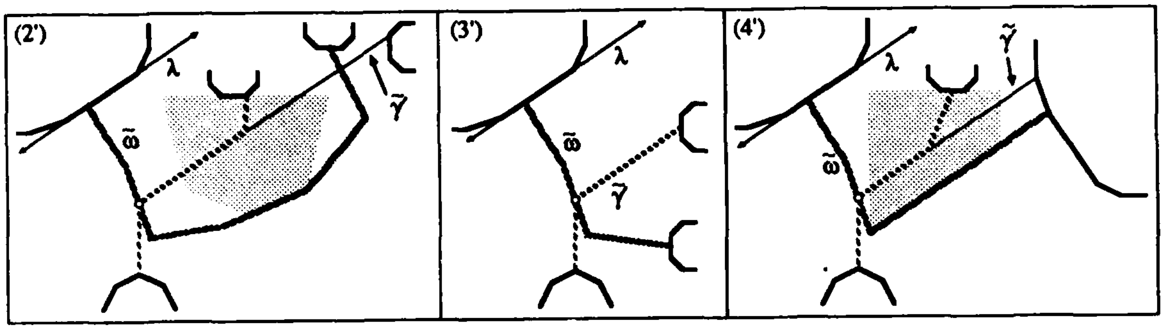


Figure 5b-3. *Straight half-cuts, continued.* The difficult case in Figure 5b-2 occurs when $\tilde{\gamma}$ and $\tilde{\omega}$ share a terminal T, and $\tilde{\omega}$ does not turn toward $\tilde{\gamma}(1)$. To handle this we replace $\tilde{\gamma}$ by a bent link $\tilde{\gamma}'$ that avoids T. Essentially the same cases arise, but $\tilde{\gamma}'$ and $\tilde{\omega}$ cannot share a second terminal T' without falling into case (4').

The remaining case is the messy one. We can assume that $\tilde{\omega}$ crosses over $\tilde{\gamma}$ at s, that (a, s) is the only crossing of $\tilde{\gamma}$ by $\tilde{\omega}$, that $\tilde{\omega}(i)$ shares a fringe with $\tilde{\gamma}(1)$, and that $\tilde{\omega}$ does not turn toward $\tilde{\gamma}(1)$ at any point in (i, s). The situation is shown in Figure 5b-3. Let F be the fringe containing $\omega(i)$, and let λ be a line tangent to F at $\gamma(1)$. Being convex, F lies on the opposite side of λ from $\gamma(0)$. Let τ be the half-cut of ω at s such that τ is parallel to λ and $\tau(0)$ lies on the opposite side of γ from $\omega(1)$. Then τ does not intersect the fringe containing $\gamma(1)$. In addition, $\sigma \star \hat{\tau}$ crosses over ω at $\omega(s)$, because otherwise $\omega_{i:s}$ would have to turn toward $\gamma(1)$.

We perform another case analysis like that above, but with $\sigma \star \hat{\tau}$ in place of γ. The details are omitted. Because ω crosses over $\sigma \star \hat{\tau}$ at s, there is no case corresponding to case (1). The remaining cases correspond to (2), (3), and (4). No further problems arise. For if ω shares one terminal with $\gamma(1)$ and the other with $\tau(0)$, geometry dictates that it must turn toward them somewhere. □

Struts for ideal routes

The following definition and proposition are central to the analysis of ideal routes. We show that wherever an ideal route turns, it has a **rigid** cut or half-cut toward which the route turns. If a cut or half-cut θ is nondegenerate and straight, and $\mathit{margin}(\theta, \Omega) = 0$, then we say θ is rigid in Ω.

Definition 5b.5. Let Ω be a design on a sheet S, and let ω route a wire in Ω. A **strut** for ω at t is a rigid cut or half-cut σ for ω at t with the following property: if $\tilde{\sigma}$ and $\tilde{\omega}$ are lifts of σ and ω satisfying $\tilde{\sigma}(1) = \tilde{\omega}(t)$, then $\tilde{\omega}$ turns toward $\tilde{\sigma}(0)$ at t. The link ω is **taut** if there is a strut for ω at every joint of ω.

The proof that ideal routes are taut is fairly intuitive. Lemmas 5b.2 and 5b.3 say that ideal routes only turn at the vertices of barriers. And since points inside barriers correspond to nondegenerate, unsafe half-cuts, it stands to reason that the vertices of barriers correspond to nondegenerate, rigid cuts and half-cuts. Using the results on chains from Section 4F, we construct a rigid half-cut for each joint of an ideal route. The lifting of this half-cut, moreover, lies in the closure of the barrier that constrains that joint. Since ideal routes turn toward the barriers that constrain them, the half-cut turns out to be a strut.

Proposition 5b.6. *Ideal routes are taut.*

Proof. Let Ω be a safe design on a sheet S, and let ω be an ideal route of a wire in Ω. Denote by M the blanket of S, and let $p\colon M \to S$ be the covering map. Lift ω to a simple link $\tilde{\omega}$ in M. Suppose ω turns at $t \in [0,1]$. Then by Lemma 5b.2 there is a barrier P for $\tilde{\omega}$ with a vertex at $\tilde{\omega}(t)$ such that $\tilde{\omega}$ turns toward the base of P at t. Let B denote the base of P, and as in Lemma 5a.4, let f denote the common flow across the half-cuts whose lifts are the forbidden half-links that define P. By the geometry of barriers (see Lemma 5a.4), there is a straight path $\tilde{\alpha}$ in $\mathit{Cl}\, P$ from B to $\tilde{\omega}(t)$ whose length is

$$\|p \circ \tilde{\alpha}\| = f + \mathit{width}(p(B))/2 + \mathit{width}(\omega)/2. \tag{5–1}$$

There is also a bent path $\tilde{\sigma}$ in $\mathit{Cl}\, P$ from $\tilde{\alpha}(0)$ to $\tilde{\omega}(t)$ whose middle lies in $P \subset M - \mathit{Bd}\, M$. Put $\sigma = p \circ \tilde{\sigma}$ and $\alpha = p \circ \tilde{\alpha}$.

The cases $t \in (0,1)$ and $t \in \{0,1\}$ have to be distinguished, but they are essentially the same. If $t \in (0,1)$, then σ is a bent half-cut for ω at t, and α is the elastic chain for σ. Also, σ is akin to a straight half-cut θ for ω (see Figure 5b-4), and hence by Proposition 5b.4, σ is semisimple (it cannot be trivial). Therefore (Proposition 4e.6) σ is nondegenerate and respects Ω. Similarly, if $t \in \{0,1\}$, then σ is a bent cut, and α is the elastic chain for σ. In addition, σ is link-homotopic to an associated cut of a straight half cut θ for ω, and again σ is nondegenerate

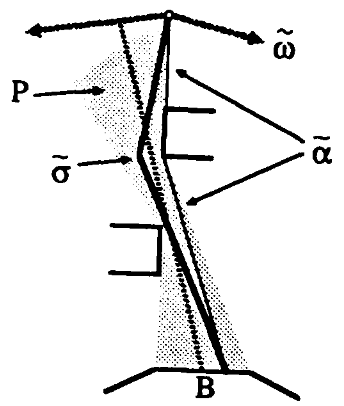


Figure 5b-4. *Finding struts for ideal routes.* This figure shows the situation at a vertex $\tilde{\omega}(t)$ for the lifting $\tilde{\omega}$ of an ideal route ω. At t the simple link $\tilde{\omega}$ turns toward a barrier P, on whose frontier $\tilde{\omega}(t)$ lies. The bent path $\tilde{\sigma}$ which ends at $\tilde{\omega}(t)$ has the straight path $\tilde{\alpha}$ as its elastic chain. The final link of $\tilde{\alpha}$ is shown to lift a strut for ω. The dashed line represents a lifting of a straight half-cut θ for ω; if σ is a half-cut, then σ and θ are akin.

and respects Ω. Also σ is nonempty. This follows from the fact that terminals are convex.

In both cases the results of Section 4F apply. Lemma 4f.3 gives us a bound on the capacity of α; together with inequality (5–1), it gives

$$\begin{aligned} cap(\alpha) &\le cap(\sigma) + \|\alpha\| - \|\sigma\| \\ &= \|\alpha\| - width(p(B))/2 - width(\omega)/2 \\ &= f = flow(\sigma, \Omega), \end{aligned} \tag{5–2}$$

and the inequality is strict if α is degenerate. Proposition 4f.1 gives us the inequality $flow(\alpha,\Omega) \ge flow(\sigma,\Omega) - gaps(\alpha)$. If $\alpha_1, \ldots, \alpha_n$ are the links of α, then we can conclude

$$\sum_{i=1}^{n} flow(\alpha_i,\Omega) \ge flow(\sigma,\Omega) - gaps(\alpha). \tag{5–3}$$

Subtracting this inequality from the equation $\sum_{i=1}^{n} cap(\alpha_i) = cap(\alpha) - gaps(\alpha)$, we get

$$\sum_{i=1}^{n} margin(\alpha_i,\Omega) \le cap(\alpha) - flow(\sigma,\Omega) \le 0. \tag{5–4}$$

The second inequality follows from (5–2). Lemma 4f.2 ensures that $n \ge 1$.

Now we deduce that the final major link in α is nondegenerate. Each α_i is a nondegenerate straight cut or half-cut for ω. Since Ω is safe and ω is evasive, all the α_i have nonnegative margin. Hence inequality (5–4) holds with equality, and therefore none of the inequalities that led up to it can be strict. In particular, $\tilde{\alpha}$ cannot be degenerate; α_n must end at $\omega(t)$. Hence α_n is a nondegenerate straight cut or half-cut τ for ω at t. By (5–4) again, $margin(\tau,\Omega) \le 0$, which implies that τ is rigid.

To check turning, let $\tilde{\tau}$ be a lift of τ satisfying $\tilde{\tau}(1) = \tilde{\omega}(t)$. To show that τ is a strut for ω at t, it remains to show that $\tilde{\omega}$ turns toward $\tilde{\tau}(0)$ at t. But this is easy, because $\tilde{\tau}(0)$ lies in $Cl\,P$. Since $\tilde{\tau}(0)$ is not on $\tilde{\omega}$, it lies on the same side of $\tilde{\omega}$ as P. And $\tilde{\omega}$ turns toward P at t by assumption. □

5C. Ideal Routes Form a Design

In this section we see the fir.st fruits of our analysis of ideal routes. We show that the ideal routes of wires in a safe design are actually wires, and that they do not intersect. Hence they actually form an embedding of the safe design, and we call this embedding an *ideal* design.

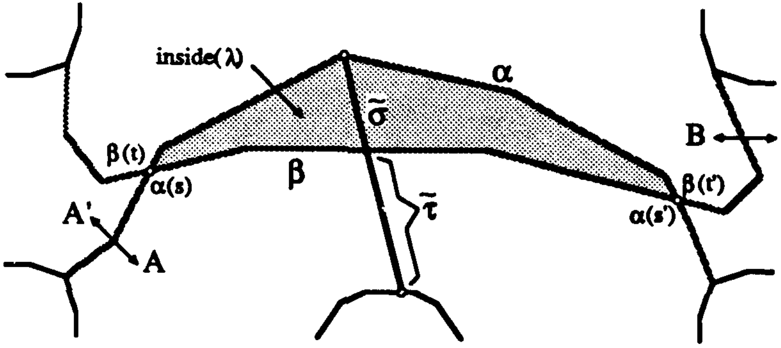


Figure 5c-1. *When lifts of ideal routes cross over.* The links α and β lift ideal routes (possibly the same one). The scraps of α are A and A'; those of β are B and B'. Where they cross over, they form a simple loop λ, and at one of its internal angles, α turns away from the endpoints of β. Since α is taut, it has a strut σ there whose lift $\tilde{\sigma}$ is shown. The half-link $\tilde{\sigma}$ crosses β, forming a half-link $\tilde{\tau}$ that ends on β. This half-link turns out to be forbidden to β.

A single technique is used both to rule out intersections between different routes and to rule out self-intersections. Assuming that two routes have an undesirable crossing, we first construct lifts of those routes that reflect this crossing. Each of these two links has its endpoints on the same side of the other. As shown in Figure 5c-1, one of the links has a joint whose strut has a lifting that crosses over the other link. We show that this strut contains an unsafe, nondegenerate half-cut for the other link. This contradicts the fact that ideal routes are evasive, and shows that the undesirable crossing could not have occurred.

The first step, finding an appropriate turning point, is handled mainly by the following lemma. Two links in a blanket **cross over** if the image of one contains points in both scraps of the other.

Lemma 5c.1. *Let α and β be coherent links in a blanket M. If α and β cross over, then there is some $z \in (0,1)$ such that, up to renaming of α and β,*
(1) *α turns away from $\beta(0)$ at z, and*
(2) *β separates $\alpha(z)$ from $\alpha(0)$.*

Proof. Let A and A' be the scraps of $M - Im\,\alpha$. By Lemma 4c.5, both endpoints of β lie in one of these scraps, say A'. Let B and B' be the scraps of $M - Im\,\beta$, and assume that B' contains the endpoints of α. The links α and β are simple because they cohere.

Suppose α and β cross over, and choose a maximal interval $(t,t') \subseteq \beta^{-1}(A)$. Define s and s' by the equations $\alpha(s) = \beta(t)$ and $\alpha(s') = \beta(t')$. Then the path $\alpha_{s:s'} \star \beta_{t':t}$ is a simple loop λ in $Cl\,A \cap Cl\,B$; the middle of $\alpha_{s:s'}$ lies in B, and the middle of $\beta_{t:t'}$ lies in A. (See Figure 5c-1.) Hence the inside of the loop λ intersects both A and B.

Corollary 3c.7 shows that λ must have at least three internal angles of measure less than π. Two of these angles can lie at $\beta(t)$ or $\beta(t')$, but the third must lie in $Mid\,\alpha_{s:s'}$ or $Mid\,\beta_{t:t'}$. If this angle is at $\alpha(x)$, where $x \in (s,s')$, then α turns toward A, and hence away from $\beta(0)$, at x. Since $\alpha(x)$ lies in B while $\alpha(0)$ lies in B', conclusions (1) and (2) hold with $z = x$. If the angle is at $\beta(y)$, where $y \in (t,t')$, then β turns toward B, and hence away from $\alpha(0)$, at y. Since $\beta(y)$ lies in A while $\beta(0)$ lies in A', conclusions (1) and (2) hold with α and β interchanged, and with $z = y$. □

A second technical lemma handles the construction of the unsafe half-cut within the strut. The strut is called σ, and the unsafe half-cut it contains is called τ.

Lemma 5c.2. *Let v and ω be ideal routes of wires in a safe design Ω. Let α and β lift v and ω, respectively, and assume $\alpha \neq \beta$. Let σ be a strut for v at z, and let $\tilde{\sigma}$ be a lifting of σ such that $\tilde{\sigma}(1) = \alpha(z)$ and α separates $\tilde{\sigma}(0)$ from the endpoints of β. Then $\tilde{\sigma}$ cannot intersect β.*

Proof. We suppose that $\tilde{\sigma}$ does intersect β and derive a contradiction. Because σ is a strut, it is nondegenerate. Let (s,b) be a crossing of $\tilde{\sigma}$ by β that minimizes s. Then $\sigma_{0:s}$ is a straight half-cut for ω at b. Call this half-cut τ. Because τ is straight and ω is ideal, τ is either trivial or semisimple, by Proposition 5b.4. By assumption, α separates the terminal of τ (which is also the terminal of σ) from the endpoints of β. Hence for τ to be trivial, its terminal would have to be a terminal of α as well, making σ trivial. But σ is nontrivial, so τ is semisimple in Ω. Proposition 4e.6 implies that τ is nondegenerate and that τ respects Ω.

Now we show that τ is unsafe, contradicting the evasiveness of ω. Because α and β cohere, Corollary 4c.4 gives us a terminal of β that is not shared by α. Suppose that $\beta(e)$ lies on this terminal, where $e \in \{0,1\}$. Let $\tilde{\chi}$ be the simple link

$\widetilde{\sigma}_{0:s} \star \beta_{b:e}$. The endpoints of $\widetilde{\chi}$ lie on opposite sides of α, and do not lie on either terminal of α. Hence $\widetilde{\chi}$ actually cuts α. If χ denotes the cut $p \circ \widetilde{\chi}$, then there is a necessary crossing (c, a) of χ by v. Applying Proposition 4d.2, we infer that

$$\mathit{flow}(\chi, \Omega) \geq \mathit{flow}(\chi_{0:c}, \Omega) + \mathit{flow}(\chi_{1:c}, \Omega) + \mathit{width}(\omega).$$

The link χ is an associated cut for τ, and since τ respects Ω, we have $\mathit{flow}(\chi, \Omega) = \mathit{flow}(\tau, \omega)$ by Lemma 4d.3. Furthermore, $\chi_{0:c}$ and σ are akin as half-cuts for v, and hence have the same flow. We conclude that

$$\mathit{flow}(\tau, \Omega) \geq \mathit{flow}(\sigma, \Omega) + \mathit{width}(\omega).$$

Since τ is shorter than σ, it follows that $\mathit{margin}(\tau, \Omega) \leq \mathit{margin}(\sigma, \Omega) - \mathit{width}(\omega)$, and the right-hand side is negative because σ is rigid. Therefore $\mathit{margin}(\tau, \Omega) < 0$, which means that τ is unsafe in Ω. □

Lemmas 5c.1 and 5c.2 are combined in the following proof.

Proposition 5c.3. *Let v and ω be ideal routes of wires in a safe design. If $v(s) = \omega(t)$, then $v = \omega$ and $s = t$.*

Proof. Let Ω be the safe design, and let M be a blanket of its sheet S with covering map $p\colon M \to S$. Suppose that $v(s) = \omega(t)$. Lift v to α and ω to β so that $\alpha(s) = \beta(t)$. Then α and β are simple. If $v \neq \omega$, then certainly $\alpha \neq \beta$; if $s \neq t$, then $\alpha(s) \neq \alpha(t)$ because α is simple, and hence $\beta(t) \neq \alpha(t)$. In both cases $\alpha \neq \beta$.

We use Lemma 5c.2 to derive a contradiction. It may be necessary to interchange v and α with ω and β, but because of the symmetry between them, we only consider the case in which no exchange is needed. By Lemma 4c.5, the endpoints of β lie on the same side of α. Let A and A' be the scraps of $M - \mathit{Im}\,\alpha$; name them so that $\beta(0) \in A'$. Let B and B' denote the scraps of $M - \mathit{Im}\,\beta$, and assume $\alpha(0) \in B'$. Suppose we find a strut σ for α at a point z, and a lift $\widetilde{\sigma}$ of σ such that $\widetilde{\sigma}(1) = \alpha(z)$ and $\widetilde{\sigma}(0) \in A$. Since the endpoints of β do not lie in A, Lemma 5c.2 will show that $\widetilde{\sigma}$ does not intersect β. There are two cases to consider.

(A) If α and β cross over, Lemma 5c.1 applies. Let z be the point given by Lemma 5c.1. By part (1), α turns away from $\beta(0)$ at z, which means α turns toward A at z. Since v is taut, there is a strut σ for v at z and a lifting $\widetilde{\sigma}$ of σ such that $\widetilde{\sigma}(1) = \alpha(z)$ and α turns toward $\widetilde{\sigma}(0)$ at z. Hence $\widetilde{\sigma}(0)$ lies in A, and Lemma 5c.2 implies that β does not intersect $\widetilde{\sigma}$. But by part (2) of Lemma 5c.1, β separates $\alpha(z)$ from $\alpha(0)$. As one can check, all the fringes of A lie in B', and hence $\alpha(0)$ and $\widetilde{\sigma}(0)$ lie on the same side of β. Thus β separates the endpoints of $\widetilde{\sigma}$, and so $\widetilde{\sigma}$ intersects β, a contradiction.

(B) Suppose instead that α and β do not cross over. Choose a maximal interval $[x, y] \subseteq \beta^{-1}(\mathit{Im}\,\alpha)$. Because α and β are simple, we have $\beta([x, y]) = \alpha([u, v])$

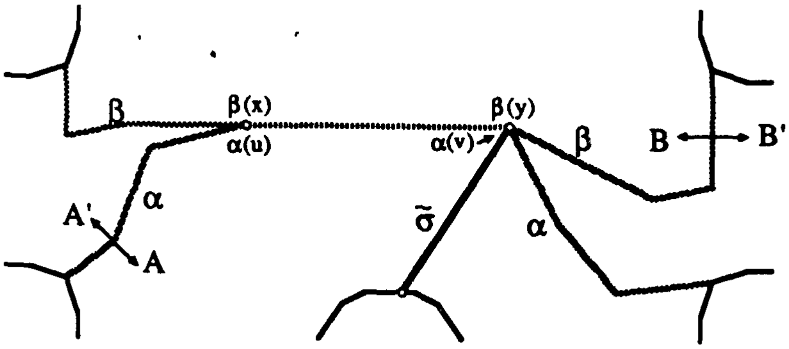


Figure 5c-2. *Intersecting lifts of ideal routes.* Figure 5c-1 does not cover the possibility that α and β intersect without crossing over. But then at some point where α and β touch, one turns away from the endpoints of the other, and essentially the same construction goes through.

for some interval $[u, v] \subset I$. (See Figure 5c-2.) There must be some point in $[u, v]$ at which α turns toward A, or some point in $[x, y]$ at which β turns toward B. By symmetry, we may assume the former; say α turns toward A at z, where $z \in [u, v]$. Because v is taut, there is a rigid half-cut σ for v at z such that if σ is lifted to $\tilde{\sigma}$ with $\tilde{\sigma}(1) = \alpha(z)$, then $\tilde{\sigma}(0) \in A$. Now Lemma 5c.2 implies that $\tilde{\sigma}$ cannot intersect $\tilde{\beta}$. But $\tilde{\sigma}$ intersects β at $\tilde{\sigma}(1)$, again giving a contradiction. □

By Lemma 5b.2 and Proposition 5c.3, the ideal routes of wires in a safe design are piecewise linear and injective, hence simple. And since they are link-homotopic to wires, their terminals are convex inner fringes. Therefore ideal routes are wires in their own right; we call them **ideal embeddings** or **ideal wires**. Proposition 5c.3 implies that the ideal wires form a design.

Corollary 5c.4. *If every wire in a safe design is replaced by an ideal route, the result is a design.* □

We call it an **ideal design**. Because the flow across a cut is the same in all embeddings of a design, as is its capacity, a cut that is safe in a design is also safe in any embedding of the design. Furthermore, a cut that is major in a design is major in any embedding of that design. Therefore ideal designs are safe.

5D. Ideal Designs Are Properly Connected

The title of this section refers to Proposition 5d.4, the main result of this section: the articles of an ideal design have disjoint extents. This proposition goes a long way toward showing that the ideal routes form a proper design, as defined in Section 4A. The most difficult part of Proposition 5d.4 is the claim that no two wires in an ideal design have overlapping extents. The method we use to prove this claim is similar to that used in Section 5C: given two ideal wires that are two close, we find a strut for one wire that gives rise to an unsafe, straight, nondegenerate half-cut for the other, contradicting the evasiveness of the second wire.

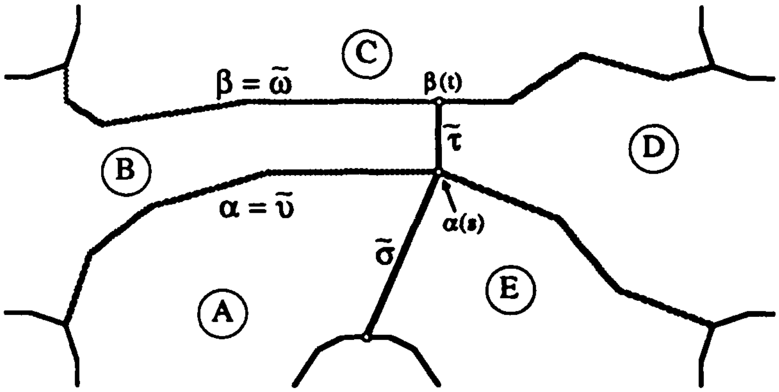


Figure 5d-1. ***When ideal wires approach too closely.*** **As in Figure 5c-1, the links α and β lift ideal routes, but this time they do not cross over. Instead, at a point of closest approach, α turns away from β. Since α is taut, it has a strut σ at this angle whose lift $\tilde{\sigma}$ is shown. The straight path $\tilde{\tau}$ lifts a minimum-length mid-cut between α and β, and the bent half-link $\tilde{\sigma} \star \tilde{\tau}$ crosses over α. Together α, β, $\tilde{\sigma}$, and $\tilde{\tau}$ split the blanket into five scraps, here denoted A through E.**

Our analysis of ideal wires continues by examining the points at which they approach each other most closely. Figure 5d-1 illuminates the situation. If two nonintersecting taut wires are not parallel, there is a point at which the wires are closest and one turns away from the other, according to Lemma 5d.1 below. Concatenating the strut for that joint with a minimum-length mid-cut between the wires, one obtains a bent half-cut for the second wire that crosses over the first wire. We prove in Lemma 5d.2 that the flow across this bent half-cut is the flow across the strut plus the width of the first wire. If the two wires have overlapping extents, then the capacity of the bent half-cut exceeds the capacity of the strut by *less than* the width of the first wire. Hence the bent half-cut is unsafe. The technical difficulties arise in proving that it is nondegenerate and that it respects the design. Lemma 4f.6 then shows that the second wire has an unsafe, straight, nondegenerate half-cut, implying that it cannot be ideal.

Turning points, again

The first step is the geometric one of finding an appropriate joint. The result we use is taken from [52].

Lemma 5d.1. *Let α and β be disjoint PL paths in R^2. There are points $\alpha(s)$ and $\beta(t)$ such that $\|\alpha(s) - \beta(t)\|$ is the minimum distance between $Im\,\alpha$ and $Im\,\beta$, and either*

(1) *α turns away from $\beta(t)$ at s; or*
(2) *β turns away from $\alpha(s)$ at t; or*
(3) *either $s \in \{0,1\}$ or $t \in \{0,1\}$.* $\square$

The proof is straightforward but messy; I refer the reader to [52].

One comment is in order about turning points in sheets and blankets. If $\widetilde{\alpha}$ lifts a link α in a sheet, and $\widetilde{\alpha}$ turns toward a point z at x, then α turns toward the projection of z at x provided that some straight path $\widetilde{\tau}$ starting at z intersects $Im\,\widetilde{\alpha}$ only at $\widetilde{\alpha}(x)$.

Construction of the bent half-cut

The bulk of the technical work is performed by the following lemma. This lemma takes care to allow the two ideal wires to coincide, because we will also need this result to prove that ideal wires are self-avoiding. We say that a subcut γ is **clean** in a design Ω if no wire in Ω intersects the middle of γ.

Lemma 5d.2. *Let v and ω be wires in an ideal design Ω, let σ be a strut for v at s, and let τ be a nondegenerate, clean, straight mid-cut between v at s and ω at t. If $\sigma \star \tau$ crosses over v at $\sigma(1)$, then $\|\tau\| \geq width(v)/2 + width(\omega)/2$.*

Proof. Let α, β, $\widetilde{\sigma}$, and $\widetilde{\tau}$ be lifts of v, ω, σ, and τ that satisfy $\widetilde{\sigma}(1) = \alpha(s) = \widetilde{\tau}(0)$ and $\beta(t) = \widetilde{\tau}(1)$. There can be no other intersections among these paths. First of all, α and β cannot cross, and neither one can intersect $Mid\,\widetilde{\tau}$ because τ is clean. Since $\widetilde{\sigma} \star \widetilde{\tau}$ crosses over α at $\widetilde{\sigma}(1)$, the link α separates β and $Mid\,\widetilde{\tau}$ from $Mid\,\widetilde{\sigma}$. And finally, α intersects $\widetilde{\sigma}$ only at $\widetilde{\sigma}(1)$ because σ is a strut for v.

We now consider the bent half-cut $\sigma \star \tau$ for ω at t. Let $\widetilde{\chi}$ denote the simple link $\widetilde{\sigma} \star \widetilde{\tau} \star \beta_{t:1}$. Its projection χ is a cut associated to $\sigma \star \tau$, and in fact $flow(\sigma \star \tau, \Omega) = flow(\chi, \Omega)$ by definition. Because τ is nondegenerate, the terminals of α and β are all distinct, and hence $\widetilde{\chi}$ cuts α. Define a by $\widetilde{\chi}(a) = \alpha(s)$. Then the crossing (a, s) of χ by v is necessary, and Proposition 4d.2 shows that

$$flow(\chi, \Omega) \geq flow(\chi_{0:a}, \Omega) + flow(\chi_{1:a}, \Omega) + width(v).$$

Now $\chi_{0:a}$ is just σ, and because σ is a strut, we have $flow(\sigma, \Omega) = cap(\sigma, \Omega)$. Denote by X the fringe containing $\sigma(0)$. Using the definition of capacity, we have

$$\begin{aligned} flow(\sigma \star \tau, \Omega) &\geq cap(\sigma, \Omega) + width(v) \\ &= \|\sigma\| - width(X)/2 + width(v)/2. \end{aligned} \tag{5–5}$$

Suppose we can prove that $\sigma \star \tau$ is safe. Then we can substitute $cap(\sigma \star \tau)$ for $flow(\sigma \star \tau, \Omega)$ in (5–5), obtaining the inequality

$$\|\sigma \star \tau\| - width(X)/2 - width(\omega)/2 \geq \|\sigma\| - width(X)/2 + width(v)/2,$$

which implies the desired result $\|\tau\| \geq width(v)/2 + width(\omega)/2$.

The next step is to prove that $\sigma \star \tau$ is nondegenerate in Ω. Let F denote the terminal of $\tilde\sigma$. If $\sigma \star \tau$ were degenerate, then F would be part of the same branch B of Ω as the terminals of β. But F and β lie on opposite sides of α. Hence that branch B would intersect α. Either B would contain the terminals of α, implying that σ and τ are degenerate, or else B would include a link η that cut α and lifted a wire of Ω. The latter is impossible, because α and η would cohere. We conclude that $\sigma \star \tau$ is nondegenerate.

To prove that $\sigma \star \tau$ is safe, we use Lemma 4f.6. It implies that if $\sigma \star \tau$ is an unsafe, nondegenerate, simple half-cut for ω that respects Ω, then ω has an unsafe, nondegenerate, straight half-cut. Since ω is evasive, the latter is false. We already know that $\sigma \star \tau$ is nondegenerate and simple, so it suffices to show that $\sigma \star \tau$ respects Ω, which is to say that its associated cuts respect Ω. By symmetry, it is enough to show that χ respects Ω. Let η be any wire in Ω, let $\tilde\eta$ and $\tilde\eta'$ be distinct lifts of η in the same branch of Ω, and suppose that $\tilde\eta$ cuts $\tilde\chi$. We must show that the terminals of $\tilde\eta'$ lie on the same side of $\tilde\chi$. I break the analysis into two cases.

(1) Suppose first that $\tilde\eta$ is not in the branch of α. It cannot be β, because β does not cut $\tilde\chi$. Hence $\tilde\eta$ cannot intersect β, α, or $\tilde\tau$ (since τ is clean). The terminals of $\tilde\eta$ must therefore be in the scraps A and E of Figure 5d-1. This means $\tilde\eta$ cuts the link $\tilde\sigma \star \alpha_{s:1}$, whose projection is an associated cut of σ, and therefore respects Ω (since σ does). Hence the terminals of $\tilde\eta'$ must lie on the same side of $\tilde\sigma \star \alpha_{s:1}$; either they both lie in A or both lie in E. In either case, they are on the same side of $\tilde\chi$.

(2) Suppose now that $\tilde\eta$ is in the branch of α. Since α cuts $\tilde\chi$, we may assume that $\tilde\eta$ *is* α. Then η' cannot intersect α, β, or $\tilde\tau$. Furthermore, since $\tilde\sigma$ respects Ω, the lift $\tilde\eta'$ cannot cut either $\tilde\sigma \star \alpha_{s:0}$ or $\tilde\sigma \star \alpha_{s:1}$. It follows that the terminals of $\tilde\eta'$ both intersect one of the five scraps A, B, C, D, and E. Hence $\tilde\eta$ does not cut $\tilde\chi$. For $\tilde\eta'$ to share a terminal with $\tilde\chi$, it would have to share a terminal either with $\tilde\sigma(0)$ or with β. Then α would be part of the branch containing either $\tilde\sigma(0)$ or a terminal of β. The former option is ruled out because $\tilde\sigma$ is nondegenerate; the latter option is ruled out because $\tilde\tau$ is nondegenerate.

Thus χ respects Ω, and the proof is complete. □

Lemma 5d.2 represents the peak of technical difficulty in the entire thesis. It brings together all the concepts we have been studying: respect, degeneracy, safety, struts, and more. There are some formidable foothills ahead, but if you have made it this far, you should be able to surmount them.

The extents of details

Proposition 5d.4 puts Lemmas 5d.1 and 5d.2 together to show that ideal wires have disjoint extents. It also shows that if two fringes in different articles have overlapping extents, then the design admits a major cut that is straight and unsafe; and if a wire's extent overlaps with that of a fringe other than one of its terminals, then the design contains a major cut or nondegenerate half-cut that is straight and unsafe. Neither of these things can happen in an ideal design. First we prove the most basic of these results.

Lemma 5d.3. *If two fringes in a design have overlapping extents, then the design admits an unsafe, straight, nonempty cut. The cut is also nondegenerate if the fringes lie in different articles.*

Proof. Let Ω be an ideal design on the sheet S. Let A and B be two different fringes of S, and suppose their extents overlap. Choose points $a \in A$ and $b \in B$ to minimize $\|a - b\|$, and σ be the straight path $a \triangleright b$. Then we have

$$\|\sigma\| < \mathit{width}(A)/2 + \mathit{width}(B)/2 \tag{5–6}$$

because the extents of A and B intersect. Neither A nor B touches $\mathit{Mid}\,\sigma$. If no other fringes of S do, then σ is the desired straight cut. It is nonempty because $A \neq B$, unsafe because inequality (5–6) implies $\mathit{cap}(\sigma) < 0$, and nondegenerate if A and B lie in different articles.

Now suppose σ intersects a fringe $C \notin \{A, B\}$. We replace σ by a shorter path τ with the same properties. By inequality (5–6), the set $\mathit{Im}\,\sigma$ is contained in the union of the extents of A and B. Hence for some $D \in \{A, B\}$ the fringe C lies within $\mathit{width}(D)/2$ units of D. Let τ be the shortest subpath of σ that runs from D to C. Then we have the analogue of inequality (5–6) for τ, namely $\|\tau\| < \mathit{width}(D)/2 + \mathit{width}(C)/2$. Moreover, we may assume that C and D lie in different articles if A and B do. For if C and A are fringes of the same article, namely the terminals of some wire, then they have the same width, and we may choose $D = B$. Similarly, if C and B fall in the same article, we may choose $D = A$. Since τ intersects fewer fringes than σ, the lemma follows by induction on this quantity. □

Now for the real result.

Proposition 5d.4. *The articles of an ideal design have disjoint extents.*

Proof. Let Ω be an ideal design on the sheet S. Let A and B be two different details of Ω, and assume they lie in different articles of Ω. We say that A and B are

too close if the distance between them (measured in the wiring norm) is less than $width(A)/2 + width(B)/2$. If A and B have overlapping extents, then A and B are too close. Supposing that A and B are too close, we derive a contradiction. Let d denote $width(A)/2 + width(B)/2$.

Case 1. Suppose that A and B are features. Then by Lemma 5d.3, the design Ω has an unsafe, straight, nonempty, nondegenerate cut. Since a nonempty and nondegenerate cut is major, this cut makes Ω unsafe.

Case 2. Let A be a feature and B a wire ω that does not touch A. Let σ be a minimum-length linear path from A to B; say $\sigma(1) = \omega(t)$. We have $\|\sigma\| < d$. We show that σ contains an unsafe straight cut or an unsafe straight half-cut for ω. In either case, Ω cannot be an ideal design. If no fringe of S touches $\sigma(1)$ or the middle of σ, then σ is a half-cut for ω at t; it is nondegenerate because A is not a terminal of ω, and it is unsafe because $\|\sigma\| < d$ implies $cap(\sigma) < 0$. Suppose instead that a fringe C touches $\sigma(1)$ or $Mid\,\sigma$. If C is a terminal of σ, then A and C are too close (because $width(C) \geq width(\omega)$) and Case 1 applies. Otherwise C is either too close to A or too close to ω, and we use the same type of induction as in Case 1.

Case 3. The interesting case is when A and B are both wires. We apply Lemma 5d.1 to these wires, call them v and ω. If some endpoint of v or ω lies within d units of the other wire, then so does the terminal containing that endpoint, and we reduce to the previous case. Otherwise, there are points $s, t \in (0,1)$ such that $\|v(s) - \omega(t)\| < d$, and either v turns away from $\omega(t)$ at s, or ω turns away from $v(s)$ at t. By symmetry, we may assume the former.

Now we apply Lemma 5d.2. Let σ be a strut for v at s; there must be one because v is taut. Let τ be the straight mid-cut from $v(s)$ to $\omega(t)$. This path does not intersect any fringe of S, or we could reduce to the previous case. Similarly, we can assume that $Mid\,\tau$ intersects no wire in Ω. Thus τ is clean in Ω, and because it connects different wires in a design, τ is nondegenerate. Finally, $\sigma \star \tau$ crosses over v at $v(s)$, because v turns away from $\tau(1)$ at s, but v turns toward $\sigma(0)$ at s. Applying Lemma 5d.2 to v, ω, σ, and τ, we see that $\|\tau\| \geq d$, contrary to assumption. □

5E. Ideal Wires Are Self-Avoiding

This section completes the proof that ideal designs are proper by showing that the wires of ideal designs are self-avoiding. The technique we use involves some fairly messy geometry, illustrated in Figure 5e-1. Beginning with a divisive article, we increase its width until just before its extent divides the sheet. At this point the frontier of its extent consists of two or more polygons linked by simple paths. One of these polygons surrounds the others, and one of the inner polygons surrounds an inner fringe of the sheet. Across one of the simple paths we find an unsafe, straight, nondegenerate subcut. If the article contains a wire, and the subcut is a cut, we prove the cut is major.

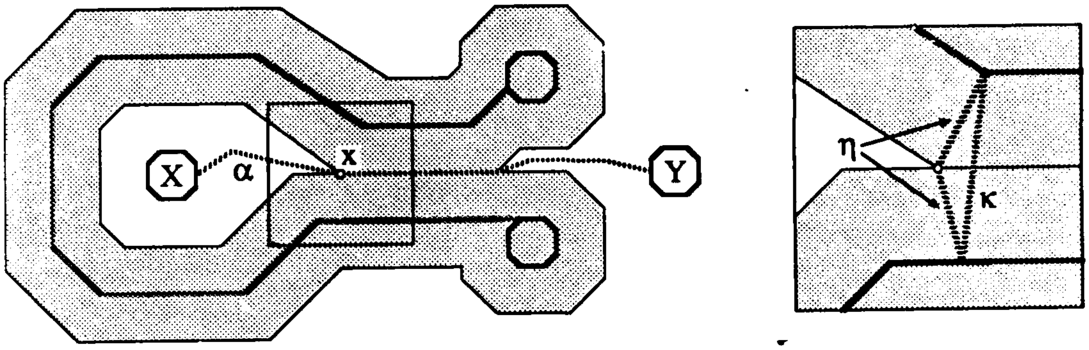


Figure 5e-1. *When an article is divisive.* **The shaded region represents a fractional extent of the article C, just before it divides the sheet by separating the fringes X and Y. It intersects no articles except C. Because this region has a vertex at x, and includes points on both sides of the path α, the dark points and line segments (at right) must contain points of C. Hence there is a bent subcut η for C which, together with C, separates X from Y. We use this fact to show that η is nondegenerate. Also η is unsafe; its capacity is negative. The straight subcut κ has the same properties.**

Fractional extents

To study self-avoidance, we adjust the widths of design details and examine the moment when an article first fails to self-avoid. Let C be any article of a design, and suppose $\delta \geq 0$. The **δ-extent** of C, denoted $T_\delta(C)$, is the extent that C would have if the widths of its details were multiplied by δ. (By convention, the 0-extent of C is the intersection of $T_\delta(C)$ for $\delta > 0$.) The set $T_\delta(C)$ is open unless $\delta = 0$. Since $T_1(C)$ is just the extent of C, the article C is self-avoiding if and only if $T_1(C)$ divides the sheet.

Given a divisive article C, we find a critical value of δ for which the δ-territory of C looks like that in Figure 5e-1. The following lemma assists the search for a critical value of δ.

Lemma 5e.1. *Let $D_0 \supseteq D_1 \supseteq D_2 \supseteq \cdots$ be a descending chain of closed, connected subsets of R^n. If $D_0 - \bigcap_i D_i$ is bounded, then $\bigcap_i D_i$ is connected.*

Proof. Set $D = \bigcap_{i=0}^{\infty} D_i$, and suppose that D is not connected. Let C and $D - C$ be nonempty sets that are both open and closed in D. Since D is closed, they are closed in R^n. Because R^n is normal, there are disjoint open sets U and V containing C and $D - C$, respectively. Write $X = D_0 - (U \cup V)$. Then X is closed, and because $X \subseteq D_0 - D$, it is also bounded. Hence X is compact. Now each of the connected sets D_i contains points of U and V, and hence must also contain points of X. It follows that the collection of closed sets $\{D_i \cap X\}$ satisfies the finite intersection condition, because if M is any finite subset of the natural numbers, it has a maximum value m, and

$$\bigcap_{i \in M} (D_i \cap X) = D_m \cap X,$$

which is nonempty. Because X is compact, the intersection $\bigcap_i (D_i \cap X)$ must be nonempty. But that intersection is precisely $D \cap X$, which is empty. This contradiction establishes the lemma. □

And the next lemma gives us a value δ with the desired properties.

Lemma 5e.2. *If an article C of an ideal design is divisive, then there exists a number $\delta \in (0, 1)$ such that $T_\delta(C)$ does not divide the sheet, but its closure does.*

Proof. Let S be the sheet. By Proposition 5d.3, the extent $T_1(C)$ of C does not intersect any fringes except those in C. For C not to self-avoid means that $T_1(C)$ divides S. On the other hand, $T_\epsilon(C)$ does not divide S for sufficiently small ϵ. Hence the quantity

$$\delta = \inf\{\, \epsilon > 0 : T_\epsilon(C) \text{ divides } S \,\}$$

is positive, and at most 1.

We shows that $\delta < 1$ by proving that $T_\delta(C)$ does not divide S. For $\epsilon < \delta$, the set $T_\epsilon(C)$ does not divide S, and hence all the fringes of S except those in C lie in a single component F_ϵ of its complement. Furthermore, for $n \geq 2$ the sets $F_{\delta - \delta/n}$ form a descending chain of connected closed sets. Call their intersection F_δ. If C is not the outer fringe of S, then the complement of F_δ is bounded; otherwise $F_{\delta/2}$ is bounded, and in either case Lemma 5e.1 applies. It shows that F_δ is connected. Since $\bigcup_{n \geq 2} T_{\delta - \delta/n}(C) = T_\delta(C)$, we have $F_\delta \subseteq R^2 - T_\delta(C)$. And since F_δ contains all the fringes of S except those in C, it follows that $T_\delta(C)$ does not divide S.

Now we indicate why $Cl\, T_\delta(C)$ divides the sheet S. Write $V = R^2 - Cl\, T_\delta(C)$. Then V is open, and because the wiring norm is polygonal, V is bounded by finitely many line segments. If all fringes except those in C lay in the same component of V, we could connect them by paths in V. The images of these paths, being compact, would lie some finite distance from $Cl\, T_\delta$. Hence they would also exist in $R^2 - T_{\delta+\epsilon}(C)$ for all sufficiently small ϵ. But by the definition of δ, the set $T_{\delta+\epsilon}(C)$ divides S for arbitrarily small positive values of ϵ. □

Deriving unsafe subcuts

Next we need a condition for a subcut to be nondegenerate. If σ is any subcut whose endpoints lie in the same article C, a **completion** of σ is any loop $\sigma \star \kappa$ where κ is a path in C.

Lemma 5e.3. *Let σ be a degenerate subcut in the design Ω. If the endpoints of σ lie in an article C of Ω, then no simple completion of σ separates two fringes that are not part of C.*

Proof. Because σ is degenerate, there is a path $\tau \in [\sigma]_P$ that lies entirely in C. This is true by definition if σ is a cut. If σ is a half-cut for ω at t, then $\sigma \star \omega_{t:1}$ is path-homotopic to a path τ in C. It follows from the groupoid properties of concatenation (Section 2A) that $\sigma \simeq_P \tau \star \omega_{1:t}$, and the right-hand path lies in C. A similar argument applies to mid-cuts.

Let $\sigma \star \kappa$ be any simple completion of σ. Then $\sigma \star \kappa$ and $\tau \star \kappa$ are path-homotopic, and enclose the same fringes. By Proposition 2c.5, the fringes enclosed by $\tau \star \kappa$ are precisely those lying inside $\sigma \star \kappa$. Now $\tau \star \kappa$ lies entirely in C. If C contains a wire of Ω, then it comprises two inner fringes of S connected by a thread. Then the other fringes of S all lie in the unbounded component of $R^2 - \mathit{Im}(\tau \star \kappa)$, whence by Proposition 2c.5, the loop $\tau \star \kappa$ does not enclose any of those fringes. Or if C is a single fringe, then $\tau \star \kappa$ may enclose C, or no fringes, or all the fringes but C (if C is the outer fringe). In no case are there fringes X and Y not in C such that $\tau \star \kappa$ encloses X but not Y. □

The last lemma of the chapter outlines the geometric construction suggested by Figure 5e-1.

Lemma 5e.4. *If C is a divisive article of an ideal design Ω, then it has a clean, straight, unsafe, nondegenerate subcut κ. Furthermore, if C includes a wire ξ, and if κ is a mid-cut between ξ at s and ξ at t, then ξ turns away from $\xi(t)$ at s.*

Proof. Let S be the sheet of Ω. First apply Lemma 5e.2 to the article C, and let $\delta \in (0, 1)$ be the quantity defined by that lemma. Write D for the open set $T_\delta(C)$. Because Ω is ideal and $D \subset T_1(C)$, Proposition 5d.4 implies that neither D nor $\mathit{Cl}\, D$ intersects any fringes of S other than those in C. Because $\mathit{Cl}\, D$ divides S, there are two fringes of S that fall in different components of $R^2 - \mathit{Cl}\, D$. Call these fringes X and Y. Since D *does not* divide S, there is a simple path α from X to Y in $R^2 - D$. (The set $R^2 - D$ is locally path-connected, being a finite union of polygons and line segments.) Clearly α must enter $\mathit{Cl}\, D - D = \mathit{Fr}\, D$. Let x be the point of R^2 where α first enters $\mathit{Fr}\, D$. Figure 5e-1 pictures the situation near x. The shaded region represents D.

We find a straight subcut κ through x. Because x lies in $\mathit{Fr}\, D$, there is a point p of C such that $\|p - x\| = \delta \cdot \mathit{width}(p)/2$, where by $\mathit{width}(p)$ we mean the width of the detail containing p. In fact, there must be points of this sort on both sides of α; call them p and q. Then $p \triangleright x$ and $q \triangleright x$ contact α from both sides, and the bent path $\eta = (p \triangleright x) \star (x \triangleright q)$ intersects C only at its endpoints; its middle lies in D. Let κ be the linear path $p \triangleright q$. By the triangle inequality we have

$$\|\kappa\| \leq \delta \cdot width(p)/2 + \delta \cdot width(q)/2. \tag{5–7}$$

I claim that *Mid* κ intersects no article of Ω. We already know that *Mid* κ is disjoint from C, and if any other article of Ω touched *Im* κ, then its extent would overlap that of C, contradicting Proposition 5d.4. So κ is a clean subcut in S, and its capacity is negative by inequality (5–7), since $\delta < 1$. Hence κ is unsafe. Furthermore, it is apparent from the geometry of Figure 5e-1 that if both endpoints of κ lie in the middle of some wire $\xi \in \Omega$, then at one of those two points, ξ turns away from the other.

It remains to prove that κ is nondegenerate. Choose a simple loop $\kappa \star \gamma$ that is a completion of κ. I claim that $\eta \star \gamma$, which is also simple loop, separates X from Y. For α crosses over $\eta \star \gamma$ at the point x and nowhere else. Since X and Y lie at the endpoints of α and do not intersect *Im* η or the article C that contains *Im* γ, they lie in different components of $R^2 - Im(\eta \star \gamma)$. Now consider $\kappa \star \gamma$. All points within the triangle Δpqx are within $\delta \cdot width(p)/2$ units of p or within $\delta \cdot width(q)/2$ units of q, and hence neither X nor Y intersects Δpqx or its inside. So $\kappa \star \gamma$ also separates X from Y, and neither X nor Y is part of C. Hence by Lemma 5e.3, the subcut κ is nondegenerate in Ω. □

Conclusions

One consequence of Lemma 5e.4 is that every divisive fringe has an unsafe, straight, nondegenerate cut. Another is that ideal wires are self-avoiding. The proof simply combines Lemma 5e.4 with Lemma 5d.2 from the preceding section.

Proposition 5e.5. *Ideal wires are self-avoiding.*

Proof. Let Ω be an ideal design on the sheet S, and let ω be a wire in Ω. Suppose ω is not self-avoiding, meaning that its article C is divisive. Apply Lemma 5e.4 to C, and let κ be the resulting subcut for ω; it is clean, straight, unsafe, and nondegenerate. There are three cases: κ can be a cut, a half-cut, or a mid-cut. If κ is a cut, then because κ is straight and terminals are convex, κ must connect the terminals of ω. Hence κ is a nonempty, and therefore major, unsafe straight cut of Ω, contradicting the safety of Ω. If κ is a half-cut, then ω is not evasive, contradicting the assumption that Ω is ideal. The remaining case is the interesting one.

Suppose that κ is a mid-cut between ω at s and ω at t. By Lemma 5e.4, ω turns away from $\omega(t)$ at s. Because ω is taut, it has a strut σ at s. We apply Lemma 5d.2 with ω representing both wires and with κ in place of τ. The conditions are easily checked: Lemma 5e.4 says that κ is clean, straight, and nondegenerate; and $\sigma \star \tau$ crosses over ω at $\sigma(1)$ because ω turns *toward* $\sigma(0)$ but *away from* $\kappa(1)$ at $\omega(s)$. The conclusion of Lemma 5d.2 is that $\|\kappa\| \geq \mathit{width}(\omega)$. But Lemma 5e.4 says that κ is unsafe, and since $\mathit{flow}(\kappa, \Omega) = 0$, this means $\mathit{cap}(\kappa) < 0$. But the capacity of κ is just $\|\kappa\| - \mathit{width}(\omega)$, which we have just shown to be nonnegative. This contradiction completes the proof. □

Together, Propositions 5e.5 and 5d.4 show that ideal designs are proper. And since every safe design has an ideal embedding, by Proposition 5b.3 and Corollary 5c.4, we obtain the following result.

Theorem 5e.6. *Every ideal design is proper, and every safe design is routable.*

Chapter 6

Routability Conditions for Designs

Chapter 5 gives conditions for a design to be routable. It shows, via the construction of an ideal embedding of a safe design, that every design whose major straight cuts are safe is routable. In addition, it provides conditions under which the fringes of a design have further desirable properties. Lemma 5d.3 shows that if all nonempty straight cuts of a design are safe, then no two fringes in the design have overlapping extents. Lemma 5e.4 implies that if all nondegenerate straight cuts of a design are safe, then no fringe in that design is divisive.

The present chapter derives converses to these results. It gives two conditions under which a design is improper: first, that it contain an unsafe major straight cut; and second, if its major straight cuts are safe, that one of its wires be shorter than its ideal embedding. These results, together with Theorem 5e.6, *imply the design* routability theorem and the design routing theorem, respectively. Section 6A also deduces the effect of an unsafe straight cut on a design. If the cut is empty but nondegenerate, then the design contains a divisive fringe. If the cut is degenerate but nonempty, then the terminals of some wire in the design have overlapping extents. Thus we obtain complete characterizations of routable designs, designs with divisive fringes, and designs with wires whose terminals are too close, in terms of the safety of straight cuts. These results are summarized in Section 6C.

Having established the design routability theorem, we proceed in Section 6D to strengthen it. Given a sheet, we find small sets of straight cuts such that, if each cut is either safe or minor in a certain design, then that design is routable. (Actually, we present techniques that would generate such sets if applied to the standard design model. We work in a slightly modifed model, and so the cut sets we find have slightly different properties.) Such a set of cuts is called *decisive* because one can decide whether a design is routable based only on the relationship of these cuts to that design. Every sheet admits a decisive cut *set whose cardinality is at most* the square of the number of fringe edges, and is possibly much smaller.

6A. Unsafe Designs Are Unroutable

The title of this section is also the content of its main result. We also prove two other useful facts here. First, every route that has an unsafe, semisimple half-cut is infeasible. Second, every design that admits an unsafe, nonempty, degenerate, simple cut includes a wire whose terminals have overlapping extents. These results are quite easy compared to those of the preceding chapter; the conceptual machinery we have built up comes through nicely.

One's intuitive picture of unsafe cuts and subcuts should resemble Figure 6a-1. If a subcut is unsafe in a design, then there is no room for the wiring in the design to fit across the it. In any embedding of the design, then, the subcut will have necessary crossings that come too close to one another or to the subcut's endpoints. In other words, one of its subpaths will be a subcut with negative capacity. If the endpoints of this subpath belong to different wires, or to a wire and a fringe other than its terminals, then these two details of the embedding have overlapping extents. If the endpoints of the subpath belong to the same wire, then the loop of wire between the two crossings must surround some fringe, or else those crossings could be removed. Consequently the extent of the wire divides the sheet, and hence that wire is not self-avoiding. If the endpoints of the subpath belong to fringes in different articles, then the extents of those articles overlap. In each case the embedded design is improper.

Of course, the actual situation is somewhat more complicated. A degenerate subcut that is not simple may have positive flow and any capacity whatsoever, and this does not mean the design is unroutable, because the endpoints and necessary crossings of this cut all involve details in the same article. For the same reason, it matters little when an empty cut is unsafe. Since it has no necessary crossings, and its endpoints lie on the same fringe, its lack of safety does not imply that any two details have intersecting extents. These technical issues are fairly easy to take into account. The results bear out our intuition: any design that contains an unsafe, major, simple cut, or an unsafe, semisimple half-cut, is improper.

Subcuts with negative capacity

As explained in Figure 6a-1, we find within an unsafe subcut a smaller subcut whose capacity is negative. If the original subcut is sufficiently nice (nondegenerate and respectful of the design), then the smaller subcut will be nondegenerate. Suppose now that its endpoints fall in a single article. Nondegeneracy means it is not path-homotopic to a path in that article. We can infer that the subcut and its article divide the sheet.

Lemma 6a.1. *Let Ω be a design on the sheet S, and let τ be a subcut with*

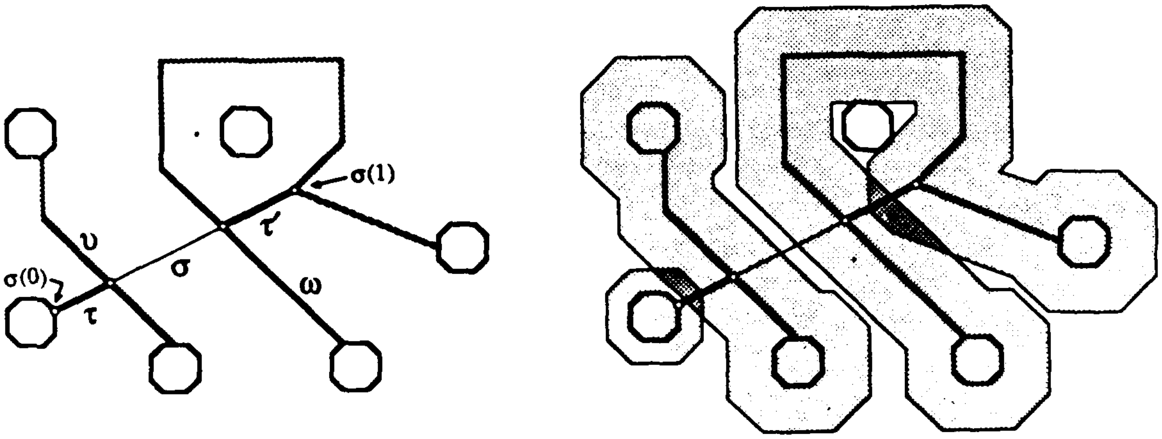


Figure 6a-1. *An unsafe half-cut.* The half-link σ shown at left is an unsafe, nondegenerate, respectful half-cut for the wire ω. The right-hand picture is the same, but it shows the extents of certain fringes and wires. These extents (shaded regions) overfill the space available in $Im\,\sigma$. The excess congestion of σ manifests itself in the nondegenerate subcuts τ and τ', which have negative capacity because they lie entirely in the extents of their endpoints. The half-cut τ shows that the wire υ is too close to the terminal of σ; the mid-cut τ' shows that ω is not self-avoiding.

endpoints in an article C of Ω. If $C \cup Im\,\tau$ does not divide S, then τ is degenerate.

Proof. Let S be the sheet, and let T denote the union of the fringes in $S - C$. Suppose T lies in a single component of $R^2 - C - Im\,\tau$. Because T and $C \cup Im\,\tau$ are compact, there is a positive distance between them. Find a simple loop λ that separates $C \cup Im\,\tau$ from T. The cases $T \subset$ *inside*(λ) and $T \subset$ *outside*(λ) are equivalent, so we assume $T \subset$ *outside*(λ).

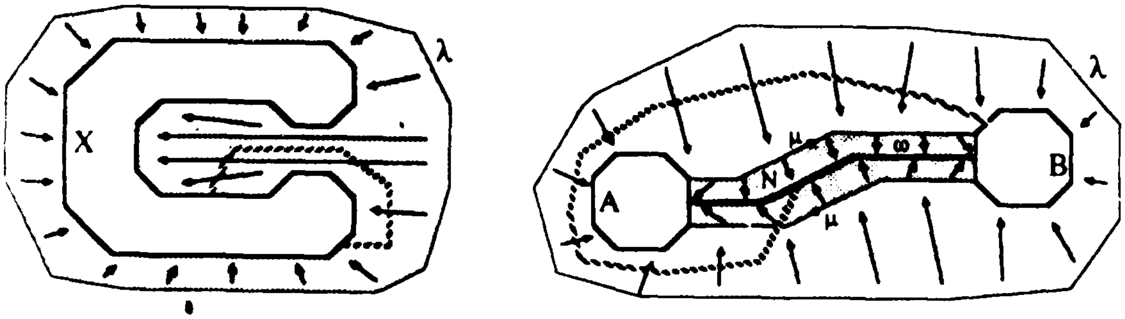


Figure 6a-2. *Retracting a degenerate subcut.* If a subcut (striped path) and its article C do not divide the sheet, then the subcut is path-homotopic to a path in C. We take the loop λ to separate C and the subcut from the other fringes. Then we construct a deformation retraction of $S \cap$ *inside*(λ) onto C. There are two cases, distinguished by whether or not C includes a wire.

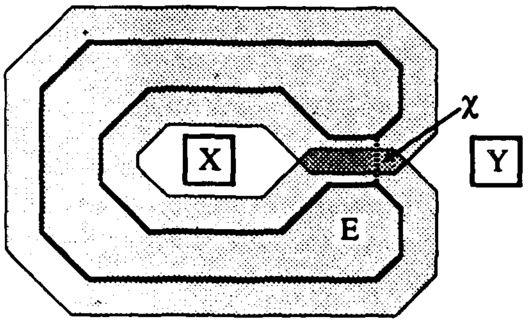


Figure 6a-3. *A nondegenerate cut whose capacity is negative.* If an empty, nondegenerate cut is unsafe, like the cut χ shown here, then the extent E of its terminal either divides the sheet (as here) or intersects a fringe in another article.

We assume the component C contains a wire ω of Ω, because this case is the harder one. Let A and B denote the terminals of ω, so that C is the article $A \cup B \cup Im\,\omega$. Construct a simple loop μ within $\textbf{\textit{inside}}(\lambda)$ such that $C \cup \textbf{\textit{inside}}(A) \cup \textbf{\textit{inside}}(B)$ is a deformation retract of the set $R^2 - \textbf{\textit{outside}}(\mu)$. Then we have

$$R^2 - \textbf{\textit{outside}}(\lambda) - \textbf{\textit{inside}}(\mu) \subseteq S,$$

and by Proposition 2c.4, this annular region has $Im\,\mu$ as a deformation retract. Combining our two deformation retractions, we obtain a deformation retraction H of the set

$$R^2 - \textbf{\textit{outside}}(\lambda) - (\textbf{\textit{inside}}(A) \cup \textbf{\textit{inside}}(B)),$$

which is $S - \textbf{\textit{outside}}(\lambda)$, onto C.

Now $S - \textbf{\textit{outside}}(\lambda)$ contains the image of τ, and the endpoints of τ lie in C. Hence the map $H(\tau(\cdot), \cdot)$ is a path homotopy between τ and a path in C. It follows that τ is degenerate. ◻

If a subcut with negative capacity has its endpoints in different articles, then as the next lemma shows, those articles have overlapping extents. Combining this fact with Lemma 6a.1, we can describe the consequences of a design having a nondegenerate subcut of negative capacity.

Lemma 6a.2. *If σ is a nondegenerate subcut in a design Ω and $\textbf{\textit{cap}}(\sigma) < 0$, then either Ω is improper or the article containing the endpoints of σ is divisive.*

Proof. Let A and B be the details of Ω that contain the endpoints of σ. We show that A and B have overlapping extents. The definition of capacity says

$$\textbf{\textit{cap}}(\sigma) = \|\sigma\| - \textbf{\textit{width}}(A)/2 - \textbf{\textit{width}}(B)/2,$$

and since $\textbf{\textit{cap}}(\sigma) < 0$, the distance between A and B, which is at most $\|\sigma\|$, is less than $\textbf{\textit{width}}(A)/2 + \textbf{\textit{width}}(B)/2$. Since the extent of A is the set of points in R^2 whose distance from A is less than $\textbf{\textit{width}}(A)/2$, and similarly for B, the extents of A and B intersect. Moreover, for every point $\sigma(t)$ we have either $\|\sigma_{0:t}\| < \textbf{\textit{width}}(A)/2$ or $\|\sigma_{t:1}\| < \textbf{\textit{width}}(B)/2$, and therefore $Im\,\sigma$ lies within the union of the extents of A and B.

If the details A and B fall in different articles of Ω, then the extents of these articles overlap, and Ω is improper. So assume the endpoints of σ lie in a single article C. Let S be the sheet of Ω. Since σ is nondegenerate, Lemma 6a.1 implies that some two fringes of S not in C are separated by $C \cup \mathit{Im}\,\sigma$. If either of these fringes intersects the extent E of C, then again Ω is improper. Otherwise they are separated by E, because E includes both C and $\mathit{Im}\,\sigma$. (Compare 6a-3, or τ' in Figure 6a-1.) This means that E divides S. In other words, C is divisive. □

One useful consequence of Lemma 6a.2 is that an unsafe, empty, nondegenerate cut in a proper design identifies its terminal as divisive. Figure 6a-3 illustrates this fact. For if an empty cut is unsafe, then its capacity is less than its flow, which is zero. And if, in addition, it is nondegenerate, then Lemma 6a.2 applies.

Unsafe subcuts have nearby crossings

The main argument of this section shows that a sufficiently nice unsafe subcut contains a nondegenerate subcut of negative capacity. And unless the original subcut is an empty cut, the smaller subcut will involve a wire. Lemma 6a.2 then allows us to conclude that the design is improper.

Proposition 6a.3. *Let σ be a nondegenerate subcut in a design Ω, and assume σ respects Ω. If σ is unsafe in Ω, then either Ω is improper or σ is empty in Ω.*

Proof. First we lift everything to the blanket. Let $\widetilde{\sigma}$ be any lift of σ. If σ is a cut, put $\widetilde{\chi} = \widetilde{\sigma}$. If σ is a half-cut for the wire ω at t, then lift ω to a simple link $\widetilde{\omega}$ with $\widetilde{\omega}(t) = \widetilde{\sigma}(1)$, and let $\widetilde{\chi} \in [\widetilde{\sigma} \star \widetilde{\omega}_{t:1}]$ be a simple link. If σ is a mid-cut between v at s and ω at t, then let $\widetilde{v}$ and $\widetilde{\omega}$ be lifts of v and ω satisfying $\widetilde{v}(s) = \widetilde{\sigma}(0)$ and $\widetilde{\omega}(t) = \widetilde{\sigma}(1)$, and let $\widetilde{\chi} \in [\widetilde{v}_{1:s} \star \widetilde{\sigma} \star \widetilde{\omega}_{t:1}]$ be a simple link. The projection $\widetilde{\chi}$ of χ is an associated cut of σ, and we have $\mathit{flow}(\sigma, \Omega) = \mathit{flow}(\chi, \Omega)$ by definition.

Next we examine the necessary crossings of $\widetilde{\sigma}$. For each wire η in Ω, the quantity $\mathit{wind}(\chi, \eta)$ is the number of lifts $\widetilde{\eta}$ of η that cut $\widetilde{\chi}$. Let $\widetilde{\eta}$ be such a lift. Both $\widetilde{v}$ and $\widetilde{\omega}$, if they are defined, share terminals with $\widetilde{\chi}$, so neither can equal $\widetilde{\eta}$. And because σ respects Ω, the lift $\widetilde{\eta}$ cannot fall in the same branch of Ω with the endpoints of $\widetilde{\sigma}$. In particular, $\widetilde{\eta}$ does not cross either $\widetilde{v}$ or $\widetilde{\omega}$, and hence crosses $\widetilde{\sigma}$. Choose for $\widetilde{\eta}$ a crossing with $\widetilde{\sigma}$. Again because σ respects Ω, two lifts of wires in Ω, if they both cut $\widetilde{\chi}$, lie in different branches of Ω. Hence all these crossings of $\widetilde{\sigma}$ fall in different branches of Ω. Each crossing has an associated width, namely $\mathit{width}(\eta)$, and the sum of these widths is $\mathit{flow}(\chi, \Omega)$ by definition.

Now we show that two crossings or endpoints of $\widetilde{\sigma}$ occur nearby. To each crossing and endpoint we associate a section of $\mathit{Im}\,\widetilde{\sigma}$, or equivalently, of $\mathit{Im}\,\sigma$. Let D_0 and D_1 denote the details of Ω containing $\sigma(0)$ and $\sigma(1)$, respectively. For $e \in \{0, 1\}$, we assign to $\sigma(e)$ the set of points $\sigma(x)$ such that $\|\sigma_{e:x}\| < \mathit{width}(D_e)/2$. If (c, r) is a

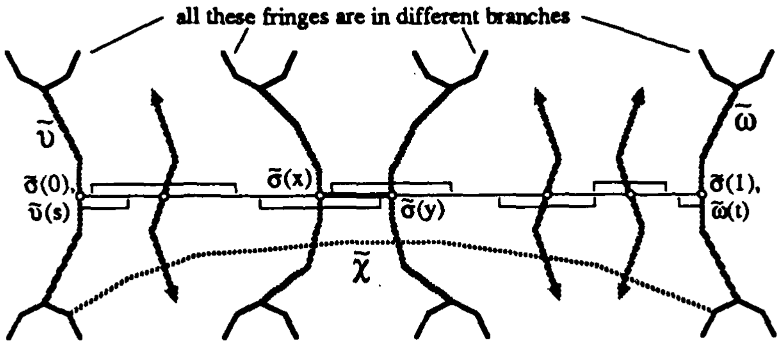


Figure 6a-4. *Contributions to the flow across a respectful subcut.* Here $\tilde{\sigma}$ lifts a mid-cut σ between υ at s and ω at t, and $\tilde{\upsilon}$ and $\tilde{\omega}$ are lifts of υ and ω that reflect the crossings $(0, s)$ and $(1, t)$. The links that cut $\tilde{\chi}$ are the liftings of wires in the design Ω that contribute to the flow across σ. They all cross $\tilde{\sigma}$, and together with $\tilde{\upsilon}$ and $\tilde{\omega}$, no two lie in the same branch of Ω. To each crossing and endpoint of $\tilde{\sigma}$ we allocate a section of $Im\,\tilde{\sigma}$ according to the width of that wire. When two of these points have overlapping sections, the subpath of $\tilde{\sigma}$ between them (shown here as $\tilde{\sigma}_{x:y}$) lifts a nondegenerate subcut of negative capacity.

crossing of σ by a lift $\tilde{\eta}$ of η that cuts $\tilde{\chi}$, then we assign to $\sigma(c)$ the set of points $\sigma(x)$ such that $\|\sigma_{c:x}\| < \mathit{width}(\eta)/2$. The combined length of all these sections of σ is

$$L = \mathit{flow}(\chi, \Omega) + \mathit{width}(D_0)/2 + \mathit{width}(D_1)/2.$$

Because σ is unsafe, we have $\mathit{flow}(\chi, \Omega) = \mathit{flow}(\sigma, \Omega) > \mathit{cap}(\sigma, \Omega)$, and hence

$$L > \mathit{cap}(\sigma, \Omega) + \mathit{width}(D_0)/2 + \mathit{width}(D_1)/2 = \|\sigma\|.$$

Therefore two of the sections of σ overlap: there are details X and Y of Ω, containing $\sigma(x)$ and $\sigma(y)$ respectively, such that $\|\sigma_{x:y}\| < \mathit{width}(X)/2 + \mathit{width}(Y)/2$. Moreover, the points $\tilde{\sigma}(x)$ and $\tilde{\sigma}(y)$ lie in different branches of Ω. Unless σ has zero flow in Ω, we can also assume that $\sigma(x)$ and $\sigma(y)$ are not both endpoints of σ.

We conclude that $\sigma_{x:y}$ is a nondegenerate subcut of Ω, and that $\mathit{cap}(\sigma_{x:y}) < 0$. Applying Lemma 6f.2, it follows either that Ω is improper, or that $\sigma(x)$ and $\sigma(y)$ lie in the same article C of Ω, and that the article C is divisive. In the latter case, Ω is still improper unless C is a fringe. And this implies that $\sigma_{x:y}$ is a cut with one terminal, whence $\sigma(x)$ and $\sigma(y)$ are the endpoints of σ, and thence $\mathit{flow}(\sigma, \Omega) = 0$. Therefore σ is an empty cut, or Ω is improper. □

Two corollaries follow immediately from Proposition 6a.3. First, we apply the proposition to semisimple half-cuts. The result justifies our concern with evasive and ideal wires.

Corollary 6a.4. *Let v embed a wire ω in a design Ω. If v has a half-cut that is unsafe and semisimple in Ω, then v is not a feasible route of ω.*

Proof. Let σ be an unsafe semisimple half-cut for v at t, and let $\Upsilon \ni v$ be an embedding of Ω. Then σ is also a semisimple half-cut in Υ, and $\mathit{flow}(\sigma, \Upsilon) = \mathit{flow}(\sigma, \Omega)$. Hence σ is unsafe in Υ. Because σ is semisimple in Υ, Proposition 4e.6 proves that σ is nondegenerate and respects Υ. Now Proposition 6a.3 applied to σ shows that the design Υ is improper. Thus v is not a feasible embedding of ω. □

Second, we apply Proposition 6a.3 to major simple cuts. The result is the easy direction of the design routability theorem.

Theorem 6a.5. *Every unsafe design is unroutable.*

Proof. Let Ω be an unsafe design. Then Ω has an unsafe major straight cut χ. Being major, χ is nondegenerate and nonempty in Ω. Let Υ be any embedding of Ω. Since flow, degeneracy, and emptiness are unaffected by link homotopy, χ is an unsafe, nondegenerate, nonempty cut for Υ. Because χ is straight, it respects Υ by Proposition 4e.2. Now apply Proposition 6a.3 to χ and Υ. It says that either Υ is improper or else χ is empty in Υ. Since χ is nonempty, Υ must be improper. Thus every embedding of Ω is improper, which means that Ω is unroutable. □

Degenerate cuts

For completeness, we consider the effects of degenerate cuts, as well as nondegenerate cuts, on the properties of a design.

Lemma 6a.6. *A design that contains an unsafe, nonempty, degenerate, simple cut includes a wire whose terminals have overlapping extents.*

Proof. Let Ω be a design, and let χ be a cut with the listed properties. Because χ is simple and degenerate ir the design Ω, Lemma 4e.3 implies $\mathit{flow}(\chi, \Omega) = 0$. And since χ is nonempty in Ω, this means χ has two terminals. These terminals lie in the same article because χ is degenerate. Therefore they are the terminals of a wire in Ω; call them A and B. Since χ is unsafe, we have

$$0 = \mathit{flow}(\chi, \Omega) > \mathit{cap}(\chi, \Omega) = \|\chi\| - \mathit{width}(A)/2 - \mathit{width}(B)/2.$$

Thus $\|\chi\| < \mathit{width}(A)/2 + \mathit{width}(B)/2$, and it follows that A and B have overlapping extents. □

6B. Ideal Embeddings Have Optimal Length

Theorem 5e.6, which says that ideal designs are proper, established the first part of the design routing theorem. This section proves the remaining part: among the proper embeddings of a routable design, the ideal design is best, in the sense that it minimizes the length of every wire. Let Ω be a safe design, and let ω be a wire in Ω. Recall that an embedding υ of ω is feasible if υ is part of a proper embedding Υ of Ω. By Proposition 5a.9, ω has an ideal embedding ρ; by Proposition 5c.4, ρ is part of an ideal design that is an embedding of Ω. And by Theorem 5e.6, this ideal design is proper. Therefore ρ is a feasible embedding of ω. We prove that no feasible embedding of ω has smaller euclidean arc length than ρ.

The struts for an ideal wire

As usual, we will study embeddings of ω by lifting them. Let S be the sheet of Ω, and let M be its blanket. Lift ω to any link $\widetilde{\omega}$, and let $Z \subset M$ be the union of the forbidden zones for $\widetilde{\omega}$. Let $\widetilde{\rho} \in [\widetilde{\omega}]_L$ be a lift of ρ. Because ideal embeddings are evasive, we have $Im\, \widetilde{\rho} \subseteq M - Z$. Let υ be any feasible embedding of ω and let $\widetilde{\upsilon} \in [\widetilde{\omega}]_L$ be a lift of υ. Then $\widetilde{\upsilon}$ has the following property.

Claim 6b.1. *If σ is a strut for ρ at s, and $\widetilde{\sigma}$ is a lift of σ satisfying $\widetilde{\sigma}(1) = \widetilde{\rho}(s)$, then $\widetilde{\upsilon}$ cannot intersect $\widetilde{\sigma}(x)$ for any $x \in [0,1)$.*

(Actually, σ can be any rigid cut or half-cut for ρ.)

Proof. Because struts are nondegenerate, the link $\widetilde{\upsilon}$ cannot intersect $\widetilde{\sigma}(0)$, for $\widetilde{\sigma}(0)$ lies on a fringe that is not a terminal of $\widetilde{\upsilon}$. Supposing that $\widetilde{\upsilon}(t) = \widetilde{\sigma}(x)$ for some $x \in (0,1)$, we prove that υ has an unsafe semisimple half-cut. If it does, Corollary 6a.4 shows that υ is not a feasible embedding of ω.

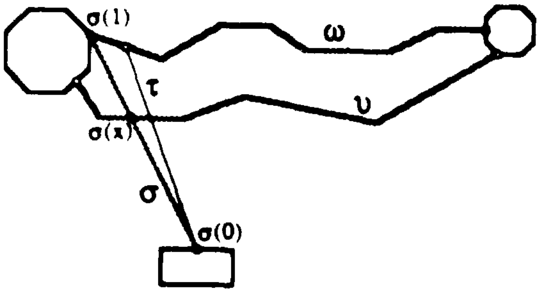


Figure 6b-1. *Sublinks of struts are semisimple.* Here the cut σ is a strut for the ideal wire ω. The half-cut $\sigma_{0:x}$ for the wire $\upsilon \in [\omega]_L$ is akin to the straight half-cut τ for ω, and hence is semisimple.

Suppose first that s is not 0 or 1. Then $\sigma_{0:x}$ is a half-cut for υ at t that is akin to, but shorter than, the rigid half-cut σ for ρ at s. Hence $\sigma_{0:x}$ is an unsafe half-cut for $\widetilde{\upsilon}$. Furthermore, since σ is semisimple (Proposition 5b.4), so is $\sigma_{0:x}$. Now suppose that s is 0 or 1. Then $\sigma_{0:x}$ is a half-cut for υ at t that has σ as an associated cut. A slight modification of σ (see Figure 6b-1) produces a straight half-cut τ for ρ which therefore is semisimple. It follows that $\sigma_{0,x}$ is semisimple, and so its associated cut τ respects Ω. Therefore the flow across $\sigma_{0,x}$ is the flow across σ, by Lemma 4d.3. Since τ is marginal and $\sigma_{0,x}$ is even shorter, $\sigma_{0,x}$ is unsafe. This completes the proof. □

We can recast Claim 6b.1 as follows. Let Σ be a set of struts for ρ, one for each point at which ρ turns, and let $\tilde{\Sigma}$ be the appropriate lifts of the struts in Σ. Define a subset X of M by

$$X = \bigcup_{\tilde{\sigma}\in\tilde{\Sigma}} \tilde{\sigma}([0,1)).$$

Then for any feasible embedding of ω, its lift $\tilde{v} \in [\tilde{\omega}]_L$ satisfies $Im\,\tilde{v} \subset M - X$. For any point t at which ρ turns, the set X contains points arbitrarily close to $\tilde{\rho}(t)$ [illegible] toward which $\tilde{\rho}$ turns at t.

Shrinking a feasible wire

[illegible] a lifting [illegible] of a feasible embedding is different from the lifting $\tilde{\rho}$ of [illegible] embedding, then we can shrink it down to $\tilde{\rho}$ without letting it touch X. [illegible] the proof of Theorem 6b.2, the main result of this section. [illegible] feasible embedding of a wire in a safe design, we apply a sequence of [illegible] that do not increase its length. Eventually it coincides with the [illegible]. The details of the proof are reminiscent of Lemmas 3d.5 and 3d.6, [illegible] somewhat more involved.

Theorem 6b.2. Let ρ be an ideal embedding of a wire ω in a safe design. Then [illegible] Euclidean arc length among all feasible embeddings of ω.

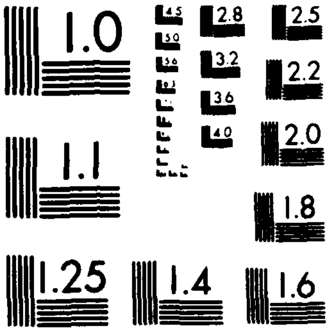
Proof. We suppose that $v \neq \rho$ is a feasible embedding of ω and find a transformation [illegible] that moves it "closer" to $\tilde{\rho}$ without increasing its length. It [illegible] that a finite sequence of these transformations takes $\tilde{v}$ to $\tilde{\rho}$. The transformations will be of two types: replacing a subpath of $\tilde{v}$ by a straight path, [illegible] parallel to itself until it coincides with $\tilde{\rho}$. At each stage the path $\tilde{v}$ [illegible] condition

$$\text{[illegible] a simple path in } M - X \text{ that is link-homotopic to } \tilde{\omega}. \tag{$*$}$$

[illegible] we may assume without loss of generality that v is canonical. [illegible] that $Im\,v$ and $Im\,\rho$ differ. For suppose $Im\,\tilde{v} = Im\,\tilde{\rho} = L$. Then [illegible] because these are the only points at which L intersects [illegible]. And since v and ρ are simple, their arc lengths are both equal [illegible] lengths of the line segments that make up L. Call this length [illegible]. Because $v \neq \rho$ by assumption, at some point $\tilde{v}(s) = \tilde{\rho}(t)$ where $s \neq t$. Assume

AD-A186 990 SINGLE-LAYER WIRE ROUTING(U) MASSACHUSETTS INST OF TECH 3/4
CAMBRIDGE LAB FOR COMPUTER SCIENCE F M MALEY AUG 87
MIT/LCS/TR-403 N00014-80-C-0622

UNCLASSIFIED F/G 9/1 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

$s < t$. Because $\tilde{v}$ and $\tilde{\rho}$ are canonical, $\tilde{v}_{0:s} \star \tilde{\rho}_{t:1}$ is a path from $\tilde{\rho}(0)$ to $\tilde{\rho}(1)$ whose length is $(1 + s - t) \cdot l$, and therefore less than l. And because its image is that of $\tilde{\rho}$, the route $v_{0:s} \star \rho_{t:1}$ is evasive. This contradicts the assumption that ρ is ideal. Therefore $Im\,\tilde{\rho}$ and $Im\,\tilde{v}$ differ, and consequently $\tilde{\rho}^{-1}(Im\,\tilde{v})$ is not all of I. We discriminate on the number of intervals of $\tilde{\rho}^{-1}(Im\,\tilde{v})$.

Case 1. Suppose first that $\tilde{\rho}^{-1}(Im\,\tilde{v})$ consists of two or more intervals. Then there are distinct crossings (a, s) and (b, t) of $\tilde{\rho}$ by $\tilde{v}$ such that the paths $\tilde{\rho}_{a:b}$ and $\tilde{v}_{s:t}$ intersect only at their endpoints. We may also assume that the middle of $\tilde{\rho}_{a:b}$ is free of crossings by $\tilde{v}$. Let λ be the simple loop $\tilde{\rho}_{a:b} \star \tilde{v}_{t:s}$. By Lemma 3c.5, the inside component N of $M - Im\,\lambda$ contains no fringes. Hence it contains no points of X, because every connected component of X includes a point on a fringe. Thus if $\tilde{\rho}$ turns at a point $c \in (a, b)$, it turns away from N.

If $\tilde{\rho}_{a:b}$ is straight, we can simply replace $\tilde{v}_{s:t}$ by the shorter path $\tilde{\rho}_{a:b}$, and $\tilde{v}$ will still satisfy condition $(*)$. Otherwise, since $\tilde{\rho}$ is piecewise straight, it turns at some point c in (a, b). Choose c so that $\tilde{\rho}_{a:c}$ is straight, and extend this path into N. Eventually it must leave N, and it cannot do so by intersecting $\tilde{\rho}_{a:b}$ since this path turns only away from N. Hence there is a straight path γ from $\tilde{v}(s)$ to some point $\tilde{v}(x)$ with $x \in (s, t)$, and $Mid\,\gamma \subset N$. Replacing $\tilde{v}_{s:x}$ by γ, we make $\tilde{v}$ shorter while preserving $(*)$.

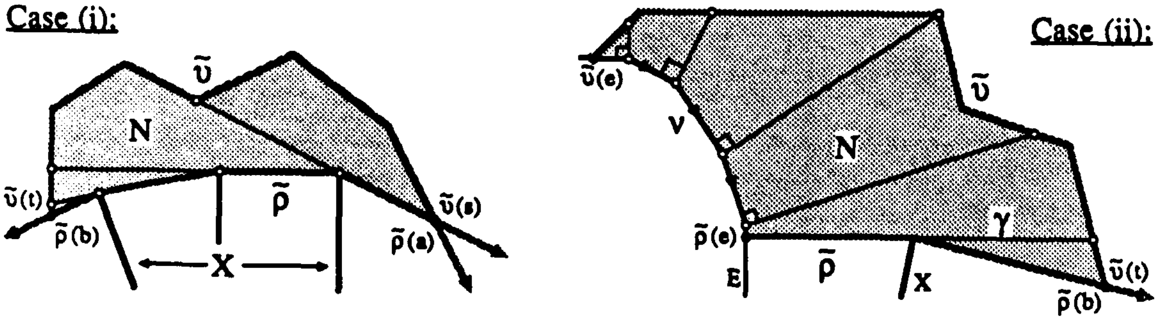


Figure 6b-2. *Ideal wires are optimal: cases 1 and 2.* At left, the lifting $\tilde{\rho}$ of the ideal embedding and the lifting $\tilde{v}$ of the other feasible embedding form a simple loop. The inside N of this loop contains none of the forbidden points X, and $\tilde{\rho}_{a:b}$ turns only away from N. Hence we can shrink $\tilde{v}$ by taking shortcuts through N (thin lines). At right, $\tilde{\rho}$ and $\tilde{v}$ form a simple loop with the path ν along the fringe E. Neither $\tilde{\rho}$ nor ν turns toward the inside N of this loop. Here we shrink $\tilde{v}$ via the shortcut γ. We construct straight paths perpendicular to ν (thin lines) to show that γ is shorter than the corresponding subpath of $\tilde{v}$.

Case 2. Suppose that $\tilde{\rho}^{-1}(Im\,\tilde{v})$ consists of only one interval. Then there is a crossing (b, t) of $\tilde{\rho}$ by $\tilde{v}$ and a point $e \in \{0, 1\}$ such that the paths $\tilde{\rho}_{e:b}$ and $\tilde{v}_{e:}$.

intersect only at $\widetilde{\rho}(b) = \widetilde{v}(t)$. We may assume that the middle of $\widetilde{\rho}_{e:b}$ is free of crossings by $\widetilde{v}$. Then the set $Im\,\widetilde{\rho}_{e:b} \cup Im\,\widetilde{v}_{e:t}$ is a web of one thread, and hence its inside N intersects no fringes except the terminal E of $\widetilde{\omega}$ that contains $\widetilde{\omega}(e)$. As in case 1, it follows that $Cl\,N \subset M - X$, and also that whenever the path $\widetilde{\rho}$ turns at some point $c \in [e, b)$, it turns away from N at c. Let ν be a simple path in E from $\widetilde{v}(e)$ to $\widetilde{\rho}(e)$. Then because the projection of E to S is a convex fringe of S, the path ν only turns away from N.

Choose a point $c \in (e, b)$ such that the path $\widetilde{\rho}_{e:c}$ is straight. When this path is extended, it must eventually leave $Cl\,N$. Because ν and $\widetilde{\rho}_{e:b}$ only turn away from N, it can only do so by intersecting $\widetilde{v}_{e:t}$. Hence there is a straight path γ in $Cl\,N$ from $\widetilde{\rho}(e)$ to some point $\widetilde{v}(x)$ with $x \in (e, t)$. We can assume that γ intersects $\widetilde{v}_{e:t}$ only at $\gamma(1)$. Replacing $\widetilde{v}_{e:x}$ by γ, we obtain a new link $\widetilde{v}$ satisfying $(*)$.

Now we show that γ is shorter than $\widetilde{v}_{e:x}$. Suppose first than ν is straight. Because $\widetilde{\rho}_{e:b}$ does not turn toward N, the angle formed by $\nu \star \widetilde{\rho}$ at $\widetilde{\rho}(e)$ is not acute. Projecting to the sheet, the result follows by elementary geometry. Suppose instead that ν is not straight. Let ν_0 be the first segment of ν, and construct a linear path that extends into N from $\nu_0(1)$, making a right angle with ν_0. This path must eventually intersect $\widetilde{v}_{e:x}$ at some point $\widetilde{v}(y)$. Let τ be the linear path from $\nu_0(1)$ to $\widetilde{v}(y)$. Again by plane geometry, τ is shorter than $\widetilde{v}_{e:y}$, so we replace the latter by the former, and replace ν by a simple path from $\tau(0)$ to $\widetilde{\rho}(e)$. Continuing in this way, we eventually reduce to the case where ν is straight.

Case 3. Suppose $\widetilde{\rho}$ and $\widetilde{v}$ do not intersect at all. Then $Im\,\widetilde{\rho} \cup Im\,\widetilde{v}$ is a web of two threads, and so its inside N contains no fringes. As in case 1, it follows that $Cl\,N \subset M - X$ and that whenever $\widetilde{\rho}$ turns at $c \in [0, 1]$, it turns away from N at c. Let E and F be the fringes of M that contain $\widetilde{\rho}(0)$ and $\widetilde{\rho}(1)$, respectively. Pick a point c such that $\widetilde{\rho}_{0:c}$ is straight, and extend it as a linear path γ until reaching $Fr\,N$. We must have $\gamma(1) \in F$ or $\gamma(1) = \widetilde{v}(x)$ for some x. If the latter, then γ is shorter than $\widetilde{v}_{0:x}$, and we proceed as in case 2. So assume $\gamma(1) \in F$. If $\gamma \neq \rho$, then by argument like that in case 2, γ is longer than $\widetilde{\rho}$. So it suffices to prove that $\widetilde{v}$ is at least as long as γ.

Let R be the scrap of $M - Im\,\gamma$ that contains $\widetilde{v}$. Construct rays β_0 and β_1 in R from $\gamma(0)$ and $\gamma(1)$, respectively, that are tangent to the fringes E and $\widetilde{F}$. Because E and F project to convex fringes, they are convex toward N at every vertex. Hence at the points where β_0 and β_1 leave $Cl\,N$, they intersect $\widetilde{v}$; say $\widetilde{v}(x) \in Im\,\beta_0$ and $\widetilde{v}(y) \in Im\,\beta_1$. The angle formed by γ with β_0 is not acute, else $\widetilde{\rho}$ would turn toward N at 0. One can also check that the angle formed by γ and β_1 is not acute. By elementary geometry again, the distance between $\widetilde{v}(x)$ and $\widetilde{v}(y)$ is at least the length of γ. Therefore $\widetilde{v}$ is no shorter than γ, and case 3 is complete.

Each case reduces to the previous one, except case (1), which applies only finitely

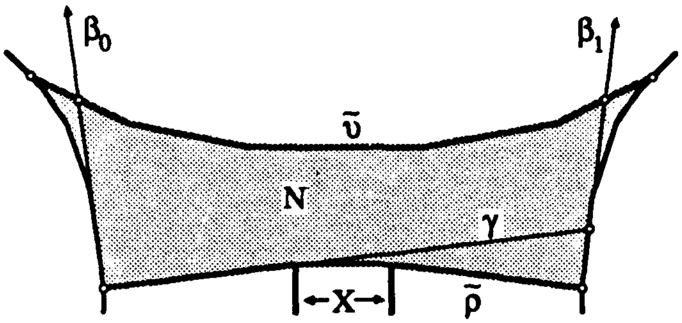


Figure 6b-3. *Ideal wires are optimal: case 3.* Here the liftings $\tilde{\rho}$ of the ideal wire and $\tilde{\upsilon}$ of the other feasible wire do not intersect. Their images form a web of two threads, and $\tilde{\rho}$ can only turn away from the inside N of this web. If γ intersects $\tilde{\upsilon}$, it shortens $\tilde{\upsilon}$, as in case 2. Otherwise, because the terminals of γ lift convex fringes, they make obtuse angles with γ on the side containing $\tilde{\upsilon}$. Hence the portion of $\tilde{\upsilon}$ between the paths β_0 and β_1 at at least as long as γ, which in turn is no shorter than $\tilde{\rho}$.

many times. Hence any feasible embedding υ of ω has a lift $\tilde{\upsilon}$ that can be reduced to $\tilde{\rho}$ by a sequence of transformations, each of which reduces or preserves the length of $\tilde{\upsilon}$. Since the length of a path in M is by definition the length of its projection to S, *this shows that ρ has minimum length among all feasible embeddings of ω.* □

Uniqueness of ideal embeddings

The proof of Theorem 6b.2 also allows us to characterize the situations in which ρ is the *unique* minimum-length feasible embedding. In cases 1 and 2, the transformation applied to $\tilde{\upsilon}$ actually reduces its length. Hence if υ is to have the same length as ρ, its lift $\tilde{\upsilon}$ must fall into case 3, that is, it cannot intersect $\tilde{\rho}$. Furthermore, the path γ constructed in case 3 must be equal to $\tilde{\rho}$; the paths β_0 and β_1 must be perpendicular to γ; their intersections with $\tilde{\upsilon}$ must lie on terminals of $\tilde{\upsilon}$; and $\tilde{\upsilon}$ must be straight. From this we conclude that $\tilde{\rho}$ and $\tilde{\upsilon}$ are straight links that intersect their terminals perpendicularly, as shown in Figure 6b-4.

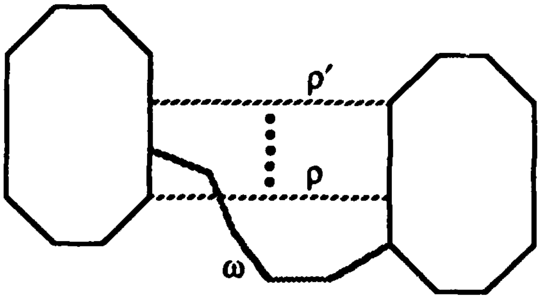


Figure 6b-4. *An ideal embedding that is not unique.* This figure shows the only situation in which a wire (ω) can have more than one ideal embedding (ρ, ρ', etc.).

6C. Summary of Design Theorems

Before moving on, we summarize the goals we have attained. This section collects the main theorems that hold in the design model and in slight modifications of that model. One of these modifications is necessary to make the sketch and design models correspond: the addition of the requirement that the terminals of a feasible wire have disjoint extents. We prove that the design routing and routability theorems continue to hold when this change is made, provided that a corresponding change is made in the definition of a safe design.

Two results stand out. First, Theorems 5e.6 and 6a.5 together characterize the routable designs.

Theorem 6c.1. (Design Routability Theorem) *Every safe design is routable, and every routable design is safe.* □

Second, Theorems 5e.6 and 6b.2 combine to characterize the optimal embedding of a safe design.

Theorem 6c.2. (Design Routing Theorem) *The ideal embeddings of the wires in a safe design form a proper design, and they have minimal euclidean arc length among all feasible embeddings of those wires.* □

Other models

Chapters 5 and 6 have been careful to consider the effects of minor unsafe cuts as well as major ones. As a result, we can now understand the effects of changing slightly the definitions of 'proper' and 'safe' designs. The results are summarized in Table 6c-1 below.

Table 6c-1 claims that the design routability theorem continues to hold in three situations. The first occurs when we strengthen the definition of a proper design to require that fringes be nondivisive, and define safe designs to be those whose *nondegenerate* straight cuts are safe. (Lemma 5e.4 shows that a design with a divisive fringe has an unsafe, nondegenerate, straight cut; Lemma 6a.2 shows that a design with an unsafe, empty, nondegenerate, straight cut has a divisive fringe, or is otherwise improper.) The second occurs when we strengthen the definition of a proper design to require that no wire's terminals have overlapping extents, and define safe designs to be those whose *nonempty* straight cuts are safe. (Lemma 5d.3 shows that a design with a wire whose terminals are too close has an unsafe, nonempty, straight cut; Lemma 6a.6 shows that a design with an unsafe, nonempty, degenerate, straight cut includes a wire whose terminals have overlapping extents.) The third situation combines the modifications of the other two.

Desired properties	Relevant cuts	Justification
Articles have disjoint extents, and wires are self-avoiding	Major straight cuts	5e.6, 6a.5
Articles are nondivisive and have disjoint extents	Nondegenerate straight cuts	5e.6, 6a.5, 5e.4, 6a.2
Wires are self-avoiding, and when two details have overlapping extents, one is a terminal of the other	Nonempty straight cuts	5e.6, 6a.5, 5d.3, 6a.6
Articles are nondivisive and have disjoint extents, and no wire has terminals with overlapping extents	Nontrivial straight cuts	All of the above

Table 6c-1. *Extensions of the design routability theorem.* A design has an embedding with the properties listed in the left column if and only if the cuts specified in the middle column are safe. The first row represents the design routability theorem itself; the second row is the most natural extension of it; and the third row describes the model that is closest to the sketch model. The fourth row represents the most restrictive model.

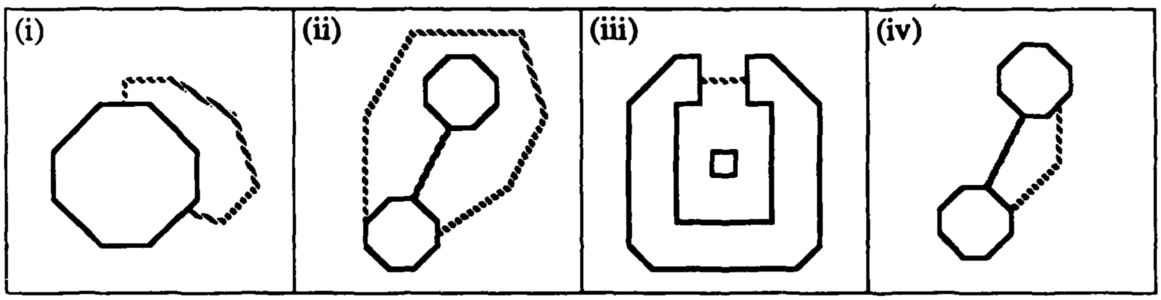


Figure 6c-2. *Four types of cuts.* Simple cuts between different articles are always relevant to routability; we assume these are safe. Other simple cuts may not be important, depending on the model. As usual, cuts are striped and wires are grey. Part (i) shows a trivial cut, which is always irrelevant. The cut in part (ii) is empty and degenerate, but not trivial; it is also unimportant. Part (iii) shows an empty but nondegenerate cut, which if unsafe may identify its terminal as divisive. The cut in part (iv) is degenerate but nonempty. If unsafe, it may indicate that the terminals of ω are too close.

Even better, the design *routing* theorem holds in all the models outlined in Table 6c-1. The reason is that all these models strengthen the conditions on the features of a proper design, but not the conditions on wires. Suppose we move from the standard design model to a more restrictive one, and strengthen the definition of safety correspondingly. Ideal designs are safe by definition, and hence their fringes will satisfy the stronger conditions. Thus ideal designs will remain proper. And since we do not weaken the conditions on feasible wires, no feasible wire in the altered model will be shorter than its ideal embedding.

Sharp safety and routability

The most important entry in Table 6c-1 is the third, which describes the model closest to the sketch model. For convenience of reference I restate that entry as a theorem. A design is **♯-proper** if its wires are self-avoiding, and when two of its details have overlapping extents, one is a terminal of the other. The design is **♯-routable** if it has a ♯-proper embedding, and **♯-safe** if its nonempty straight cuts are safe. (The terminology will seem less strange in Chapter 8.) Every ♯-safe design is safe, and hence the design routing theorem applies to all ♯-safe designs. The design routability theorem, on the other hand, becomes the following.

Theorem 6c.3. *Every ♯-safe design is ♯-routable, and every ♯-routable design is ♯-safe.* □

6D. Cuts That Decide Routability

A useful interpretation of the design routability theorem is this: For every sheet, there are certain cuts (namely, all straight cuts) whose safety, emptiness, and degeneracy in a design determine the routability of that design. I call such a set of cuts **decisive**. If every sheet has a small, easily computable, decisive set of cuts, then routability testing reduces to the problem of computing the flow across a cut and determining whether a cut is degenerate. The first problem is can be solved by the techniques of Section 7C. The second problem goes away if we want to test ♯-routability rather than routability.

Definition 6d.1. A set of cuts Γ on a sheet S is **♯-decisive** (under a particular wiring norm) if for every design Ω on S that is not ♯-routable, some cut in Γ is unsafe and nonempty in Ω.

In this section we find finite ♯-decisive sets of straight cuts. Not only are they finite, in fact, but their size is at most quadratic in the complexity of the sheet, by which I mean the number of convex polygons needed to define its boundary. (See Definition 6d.7 below.) We derive the decisive sets by successive refinement, starting from the set of all straight cuts on a given sheet. Theorem 6c.3 implies that the set of all straight cuts on a sheet is ♯-decisive. Our first result shows how one ♯-decisive set may be reduced to a smaller one.

A criterion for decisiveness

We make use of link homotopy to reduce the number of cuts in our $\sharp$-decisive sets. If two cuts are link-homotopic, then they have the same flow and emptiness in every design. Only the shorter one needs to be included in any $\sharp$-decisive set, because if the longer one is unsafe and nonempty, so is the shorter one. In any given sheet, the number of link classes of straight cuts is finite, so one might try to find a $\sharp$-decisive cut set by choosing a minimum-length cut from each link class that contains nontrivial straight cuts. Not all link classes have minimum-length elements, however.

Fortunately, when a link class $[\alpha]_L$ has no minimum-length element, there is a chain γ for a link in $[\alpha]_L$ whose length is no greater than that of any link in $[\alpha]_L$. Using the results of Section 4F, we can show that some link of γ is unsafe and nonempty whenever any link in $[\alpha]_L$ is unsafe and nonempty. Consequently the link class $[\alpha]_L$ may be discarded in favor of the link classes of the links of γ. We say α is *weak*, because other cuts give stronger constraints on $\sharp$-routability. The precise definition we need is the following.

Definition 6d.2. Let Σ and Γ be sets of linear paths, and suppose every path in Σ is a cut in the sheet S. We say Γ **dominates** Σ if for every cut $\sigma \in \Sigma$, there is a straight path γ with $\|\gamma\| \leq \|\sigma\|$ and either

(1) γ is a link in $\Gamma \cap [\sigma]_L$, or

(2) γ is a chain for a link in $[\sigma]_L$, and γ contains either two or more links or an edge of a fringe of S.

If Σ is the set of nontrivial straight cuts in S, then Γ is called **dominant** in S.

Dominance is transitive. For if a path τ dominates σ, then either $\|\tau\| \leq \|\sigma\|$ and $\tau \simeq_L \sigma$, in which case any path that dominates τ also dominates σ, or condition (2) holds for σ and some chain γ. In the latter case we say σ is **weak**. Weak cuts are dominated by every cut set, even the empty set. *Dominant* cut sets are $\sharp$-decisive, as we now prepare show. The name of the game is finding dominant cut sets.

Lemma 6d.3. *Let Γ dominate the set of nontrivial straight cuts in a sheet S. If any straight cut σ in S is unsafe and nonempty in a design Ω, then some cut $\gamma \in \Gamma$ with $\|\gamma\| \leq \|\sigma\|$ is unsafe and nonempty in Ω.*

Proof. Because σ is nonempty, it is nontrivial, and hence Γ dominates it. We show that Ω has an unsafe, nonempty cut in Γ by successively reducing σ. Let γ be the straight chain that is related to σ as in Definition 6d.2.

In case (1) the path γ itself is the unsafe cut in Γ. Case (1) says that γ is a cut that is link-homotopic to σ, so $\mathit{flow}(\gamma, \Omega) = \mathit{flow}(\sigma, \Omega)$ by Proposition 4b.3. Since $\|\gamma\| \leq \|\sigma\|$, we have $\mathit{cap}(\gamma) \leq \mathit{cap}(\sigma)$, and hence the fact that σ is unsafe in Ω implies that γ is unsafe in Ω. Similarly, γ is nonempty because σ is nonempty.

In case (2) we reduce σ to a link of γ that is still unsafe and nonempty. Let δ be the minimum distance from a fringe vertex to a fringe edge that does not contain that vertex. Then every link of a straight chain in S has length at least δ, and every edge of a fringe of S has length at least δ. Hence by the conditions on γ, the unsafe link we find will be shorter than σ by at least δ. So the process of finding shorter and shorter unsafe cuts must eventually terminate with a cut falling into case (1). If γ contains only one link λ, one can show that $\lambda \simeq_L \sigma$. Since $\|\lambda\| < \|\sigma\|$, the link λ is both unsafe and nonempty. Henceforth we suppose that γ has two or more links.

Assume first that σ is nonempty but degenerate. Because σ is straight, Corollary 4e.3 shows that $\mathit{flow}(\sigma, \Omega) = 0$. Hence the terminals of σ are the two terminals of a wire in Ω, call them A and B. Since these terminals are convex, and γ contains two or more links, γ must intersect other fringes as well. Say γ contains links from A to C and from B to D, where $A \neq C$ and $B \neq D$ but possibly $C = D$. Because $\mathit{cap}(\gamma, \Omega) < 0$, the extents of A and B overlap. Therefore either A and C have overlapping extents, or B and D do, or perhaps both. In either case γ contains a link of negative capacity between different fringes: it is unsafe and nonempty.

The remaining possibility is that σ is major. In this case, let $\alpha \in [\sigma]_L$ be a link for which γ is a chain. We may replace α by any link in $[\gamma]_P$, so choose α very close to γ in length. Specifically, let $\|\alpha\| - \|\sigma\|$ be less than $-\,\mathit{margin}(\sigma, \Omega)$. Then because $\|\gamma\| \leq \|\sigma\|$ we have

$$\begin{aligned} \mathit{cap}(\alpha) &= \mathit{cap}(\sigma) - (\|\sigma\| - \|\alpha\|) \\ &\leq \mathit{cap}(\sigma) - \mathit{margin}(\sigma, \Omega) \\ &= \mathit{flow}(\sigma, \Omega). \end{aligned}$$

Since $\mathit{flow}(\alpha, \Omega) = \mathit{flow}(\sigma, \Omega)$ by Proposition 4b.3, it follows that α is unsafe. And because γ is linear, it is the elastic chain for α (Lemma 3d.3). Let $\gamma_1, \ldots, \gamma_n$ be the major links of the chain γ. Proposition 4f.1 and Lemma 4f.3 bound the flow and capacity of these links:

$$\sum_{i=1}^{n} \mathit{cap}(\gamma_i) \leq \mathit{cap}(\alpha) - \mathit{gaps}(\gamma); \qquad \sum_{i=1}^{n} \mathit{flow}(\gamma_i) \geq \mathit{flow}(\alpha) - \mathit{gaps}(\gamma).$$

Subtracting the latter from the former gives us the inequality $\sum_{i=1}^{n} \mathit{margin}(\gamma_i, \Omega) \leq \mathit{margin}(\alpha, \Omega)$, and the left-hand side is negative. Hence γ_i is unsafe for some i. This completes the proof. □

Corollary 6d.4. *Every dominant cut set is $\sharp$-decisive.*

Proof. Suppose Γ dominates the set of nontrivial straight cuts in a sheet S. Let Ω be a design on S, and suppose Ω is not $\sharp$-routable. Because the set of all straight cuts in S is $\natural$-decisive, Ω has a unsafe, nonempty, straight cut, call it σ. Lemma 7c.3 now gives us an unsafe, nonempty cut in Γ. □

Minimal paths between fringe edges

Rather than deal with link-homotopy classes, there is a purely geometric way to find dominant cut sets. It involves choosing paths of minimum length between fringe edges. Let P and Q be two compact regions in the plane. A **minimal** path from P to Q is a linear path α from P to Q such that $\|\alpha\|$ equals $\|P - Q\|$, the minimum distance between a point of P and a point of Q. Together with Corollary 6d.4, the following result implies that a $\natural$-decisive cut set may be obtained for a sheet S by choosing a minimal cut between each pair of fringe edges of S, whenever such a cut exists.

Lemma 6d.5. *If σ is a nontrivial straight cut, and τ is a minimal path between the same fringe edges as σ, then τ dominates σ.*

Proof. Let S be the sheet in which σ is a cut. Let P be a fringe edge containing $\sigma(0)$ and $\tau(0)$, and let Q be a fringe edge containing $\sigma(1)$ and $\tau(1)$. Say P **points at** Q if the line containing P intersects the middle of Q. If the segments P and Q are parallel, then σ must be a minimal path from P to Q. In this case $\sigma = \gamma$, and we are done. We may assume P and Q are not parallel, whence at most one of them can point to the other.

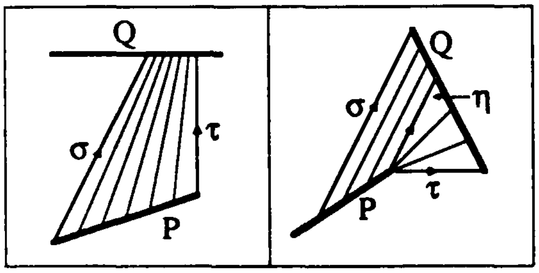


Figure 6d-1. *Reducing one linear path to a shorter one.* Each panel shows seven stages of a homotopy H between σ and τ, namely $\eta_y = H(\cdot, y)$ for $y = \frac{0}{6}, \frac{1}{6}, \ldots, \frac{6}{6}$. Every η_y is a linear path from the fringe edge P to the fringe edge Q.

We first construct a family of linear paths $\{\, \eta_y : y \in I \,\}$ that interpolate between σ and τ. These paths determine a homotopy $H: I \times I \to R^2$ by $H(x,y) = \eta_y(x)$. If neither of P and Q points at the other, then H interpolates linearly between σ and τ: we put

$$H(x, \cdot) = \sigma(x) \triangleright \tau(x). \tag{6-1}$$

Otherwise, supposing that P points at Q, we define η to be the linear path parallel to σ from $\tau(0)$ to Q, and define H by

$$H(x, \cdot) = (\sigma(x) \triangleright \eta(x)) \star (\eta(x) \triangleright \tau(x)). \tag{6-2}$$

This definition says that for $0 \le y \le \frac{1}{2}$, the path η_y moves parallel to itself from σ to η, and for $\frac{1}{2} \le y \le 1$, it pivots around $\tau(0)$ going from η to τ. In both cases $\eta_0 = \sigma$ and $\eta_1 = \tau$.

The key properties of H are that for each $y \in I$, the path η_y is linear, $\|\eta_y\| \le \|\sigma\|$, and η_y intersects P and Q at its endpoints alone. Linearity is immediate from the definitions. Now we prove that $\|\eta_y\|$ is maximal at $y = 0$. When equation (6–1) holds, the linearity of H and the convexity of $\|\cdot\|$ imply that $\|\eta_y\|$ is a convex function of y. Since τ is minimal, $\|\eta_y\|$ is a nonincreasing function of y. The same argument applies to equation (6–2), at least for $y \ge \frac{1}{2}$; for $y \le 1/2$ the path η_y is a shrunken copy of σ, and hence $\|\eta_y\|$ is nonincreasing on $[0, \frac{1}{2}]$ also. The third property, which concerns the intersection of η_y with P and Q, should be geometrically obvious.

Now we prove that either H is a link homotopy, or else some path η_y is a chain for a link in $[\sigma]_L$ and contains either two links or a fringe edge. In either case τ dominates σ. Let z be the infimum of the values y such that η_y is a link in S, and put $\gamma = \eta_y$. Note that $\|\gamma\| \le \|\sigma\|$. There are two cases.

(1) If $z = 1$ and γ is a cut, then η_y is a link for all y, and hence H is a link homotopy between σ and γ. Here γ falls into case (1) of Definition 6d.2.

(2) If γ is not a cut, then γ is a chain for a link χ in $[\sigma]_L$; one can extract homotopies from H to prove $\sigma \simeq_L \chi \simeq_P \gamma$. This could only fail if γ were constant, but then σ would be trivial, contrary to assumption. The middle of γ must contain a vertex of S, and this vertex is not on P or Q. Either this vertex is connected to $\gamma(0)$ or $\gamma(1)$ by a fringe edge, or else γ contains two or more links. Thus γ falls into case (2) of Definition 6d.2.

The remaining possibility, that $z < 1$ and $\gamma = \eta_z$ is a cut, can be ruled out. For in this case η_y is a cut for all y sufficiently close to z. □

Using Lemma 6d.5 and the transitivity of dominance, one can obtain smaller ♯-decisive cut sets. The following lemma shows that one need consider only cuts that are locally minimal, that is, minimal with respect to all the fringe edges they intersect. Formally, a linear path α in R^2 is **locally minimal** in the sheet S if there are fringe edges P and Q of S that contain $\alpha(0)$ and $\alpha(1)$, respectively, and whenever P and Q are such edges, α is a minimal path from P to Q.

Lemma 6d.6. *The nontrivial, locally minimal cuts in a sheet are dominant.*

Proof. Let σ be a nontrivial, nonweak cut in S. It suffices to prove that some locally minimal cut τ in S dominates σ. Let P and Q be fringe edges of S that contain $\sigma(0)$ and $\sigma(1)$, respectively. Choose a minimal path τ from P to Q. By Lemma 6d.5, τ dominates σ. Because σ is not weak, τ is a cut in S. If τ is not locally minimal in S, then there are fringe edges containing the endpoints of τ which are closer to one another (in the wiring norm) than P and Q. Replace P and Q by these fringe edges, replace σ by τ, and repeat. Since the number of pairs of fringe edges is finite, we must eventually find a cut τ that is locally minimal in S. By the transitivity of dominance, τ dominates σ. □

The boundary of a sheet

The final result of this section provides our strongest and most general $\sharp$-decisive cut sets. Definition 6d.7 points the way.

Definition 6d.7. An **edging** for a sheet S is a finite set Δ of convex polygons and line segments in $R^2 - (S - Bd\, S)$ whose union contains $Bd\, S$. A cut set Γ **spans** the sheet S if S has an edging Δ such that for every two elements $P, Q \in \Delta$, either Γ contains a minimal path from P to Q that is a cut in S, or else there is a minimal path from P to Q that is *not* a cut in S.

One natural way to obtain an edging for a sheet is to express its fringes as unions of polygonal obstacles. (The simplest edging for a sheet is just the set of fringe edges.) Minimal cuts between these obstacles determine a $\sharp$-decisive set.

Proposition 6d.8. *Every cut set that spans a sheet is dominant.*

Proof. Let Γ span the sheet S, and let Δ be the edging satisfying the condition of Definition 6d.7. Because dominance is transitive, it suffices by Lemma 6d.6 to show that every nontrivial, locally minimal cut α in S is dominated by Γ. We may assume that α is not weak. Choose $P, Q \in \Delta$ such that $\alpha(0) \in P$ and $\alpha(1) \in Q$, and such that both P and Q contain edges of $Bd\, S$. Because α is locally minimal in S, it is a minimal path from P to Q. Let γ be a minimal path from P to Q that is either a cut in Γ or not a cut. We prove that either

(1) γ dominates α, or

(2) there is a shorter cut α' that dominates α and is not locally minimal.

In case (2) we repeat the process, replacing P and Q by two elements of Δ that are closer. Since Δ is finite, this process must eventually terminate in case (1). By transisitivity of dominance, then, Γ dominates α.

We make use of the geometry of P and Q. Let P^* denote the set of points in P at which the minimum distance to Q is achieved, and let Q^* denote the set of points in Q at which the minimum distance to P is achieved. Because P and Q are line segments or convex polygons, both P^* and Q^* are points or line segments. If $(P^* \cup Q^*) \subseteq Bd\, S$, then α' is a path between the same fringe edges as α. In this case α' dominates α by Lemma 6d.5, and case (1) occurs. Now suppose that P^* and Q^* do not both lie in $Bd\, S$. Let A and B be the components of $P^* \cap Bd\, S_\epsilon$ and $Q^* \cap Bd\, S_\epsilon$, respectively, that contain the endpoints of α. Then either A intersects a fringe edge that approaches closer to Q^* than P^*, or B intersects a fringe edge that lies closer to P^* than Q^*. In either case there is a minimal path α' between A and B that is not locally minimal in S. This path dominates α by Lemma 7b.5, and case (2) occurs. □

Proposition 6d.8, in combination with Corollary 6d.4, will be useful for proving the sketch routability theorem and the correctness of Algorithm T.

Chapter 7

From Theory to Algorithms

Despite all the theorems of preceding chapters, proofs of correctness for the algorithms in Chapter 1 still lie some distance away. Two difficulties must be overcome: the gap between the sketch and design models, and the difference between the algorithmic constructions of Chapter 1 and the mathematical definitions of Chapters 4 and 5. For example, we must show that the ideal embedding of a wire may be constructed by merging the shortest paths through certain corridors, and that the same technique applied to a sketch produces a proper realization of that sketch.

My strategy is to justify the algorithmic techniques in the context of designs, and then carry them over to sketches. This chapter discusses methods for testing the routability of a design and computing the ideal embeddings of its wires. The techniques presented in this chapter are those embodied by the sketch algorithms of Chapter 1, adapted to the design model. Here the emphasis is on theorems, however, rather than detailed algorithms. The following chapter addresses the differences between designs and sketches, and uses the results on design algorithms to prove the correctness of the sketch algorithms.

Because the main purpose of this chapter is to explain the algorithms of Chapter 1, in preparation for proving them correct, it ignores certain issues that arise in the design model. Difficulties arise because the terminals of a wire in a design are not points. When routing a wire in a design, for instance, one must consider where its endpoints should be, whereas in a sketch the endpoints of every trace are fixed. Consequently I do not describe how to find the endpoints of a wire's ideal embedding. This omission is reasonable since the design model is not appropriate for practical use. (Section 10C notes that complete algorithms do exist for routing and testing the routability of designs. They are less efficient, however, than the corresponding algorithms for sketches.)

Chapter outline

To understand this chapter it helps to be familiar with Chapter 1, since the two share many ideas. Section 7A relates path homotopy to gate lists, and thereby lays the groundwork for algorithms that find routes for wires. Section 7C shows how the flows across cuts and half-cuts may be computed by counting crossings with elastic chains.

The final two sections concern the construction of ideal embeddings. By analyzing the composition of ideal wires, we show that the ideal embedding of a wire can be obtained by merging suitable partial embeddings of the wire, as in Algorithm R.

7A. Geometric Representations of Path Classes

The first thing one needs in an algorithm that deals with homotopy constraints is a means of working with homotopy classes. Many different representations are possible. If the paths in question are loops at a common base point, the fundamental group suffices. The representation I use is slightly more general, and is derived just as one might compute the fundamental group: by explicit construction of a blanket. It may seem odd to perform this construction after having derived so many facts about blankets, and not before. But knowing how to build a blanket would have simplified few of those results. Moreover, the mathematical results of this thesis do not depend upon this construction at all; I use it only to justify some algorithmic techniques.

My algorithms represent the path class of a path by means of its crossing sequence with a set of cuts that partition the routing region. In Chapter 1 we called these sequences *corridors*. In this section I define a similar notion for the design model, called the *path code* of a path. I then prove that the endpoints and path code of a piecewise linear path define its path class.

Patterns and seam lists

We first view the sheet as being made of simply connected *pieces* sewn together along *seams*. The blanket will be constructed by sewing together infinitely many copies of these pieces.

Definition 7a.1. Let S be a sheet, and let Γ be a finite set of disjoint simple cuts of S. Let $S_0, \ldots, S_m$ be the closures of connected components of $S - \bigcup_{\gamma \in \Gamma} Im\, \gamma$, regarded as subspaces of S. Suppose that each set S_k is simply connected, and that each cut γ_i lies in exactly two of them. Then Γ is a **pattern** for S. Its elements are called **seams**, and the sets S_k are called the **pieces** of the pattern.

Given a pattern Γ for a sheet S, we define for every PL path α in S a **seam list**. The seam list of α in Γ is a word over the alphabet Γ that records the sequence in which α crosses over the seams in Γ. Because α is PL, it can cross over the seams of Γ only finitely many times, and because the seams in Γ are disjoint, the intervals

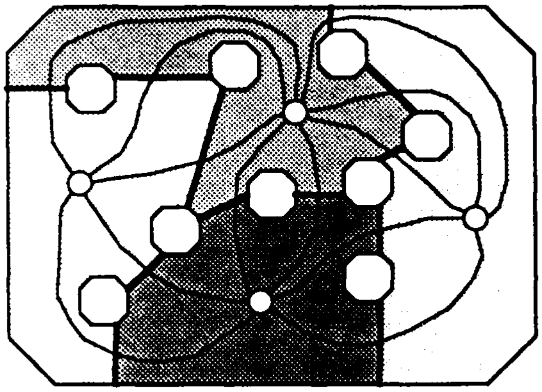


Figure 7a-1. *A pattern for a sheet, and the corresponding graph.* The seams (dark lines) separate the sheet into simply connected pieces (closures of shaded regions). The adjacency relation of these pieces forms a graph (circles and light curves) with no self-loops. The *seam list* of a path α—the sequence of seams that α crosses over—corresponds to a unique path in this graph, provided that no seam contains either endpoint of α.

in which α crosses over the various seams occur in a definite order along α. (Minor complications would arise if one allowed the seams to intersect at their endpoints, as is desirable for some applications.) In this section I write sequences of seams as strings, with a centered dot for concatenation. The empty seam list is denoted ϵ.

We represent the path class of a chain by its endpoints and its *reduced* seam list, or **path code**. Starting from the seam list of a PL path α, the reduced seam list of α is obtained by repeating the following reductions until no further reductions apply. Where two occurrences of the same seam γ_i are consecutive, delete both. If the first seam is γ_i and $\alpha(0) \in Im\,\gamma_i$, delete the first seam. If the last seam is γ_j and $\alpha(1) \in Im\,\gamma_j$, delete the last seam. These reductions evidently terminate in a unique sequence.

For almost all piecewise linear paths α in S, the path code and the endpoints of α determine the path class $[\alpha]_P$. Unfortunately, this characterization only holds for paths α that are **free** in the pattern Γ, meaning that no seam of γ contains either endpoint of α. The goal of this section is to prove Proposition 7a.8: In the presence of a pattern Γ, two free PL paths are path-homotopic if and only if they have the same endpoints and the same path code. The "if" direction is fairly straightforward. Its core is the following lemma.

Lemma 7a.2. *Let α be a PL path in a sheet S. For any pattern Γ on S, there is a PL path $\beta \in [\alpha]_P$ whose seam list in Γ is the path code of α in Γ. If α is a link, so is β, and if the seams of Γ are straight, then $|\beta| \leq |\alpha|$.*

Proof. Let $\Gamma = \{\gamma_1, \ldots, \gamma_n\}$ be a pattern on S, and let ξ denote the seam list of α in the pattern Γ. By induction, it suffices to show that if some string ζ can be obtained from ξ by a single reduction, then there is a path $\beta \in [\alpha]_P$ satisfying $|\beta| \leq |\alpha|$ whose seam list is ζ. There are three cases, one for each reduction rule. All three cases are very similar, so we consider only one case.

Suppose that ξ equals $u \cdot \gamma_i \cdot \gamma_i \cdot v$ for some substrings u and v, and that ζ is $u \cdot v$. Then there are points $s, t \in \alpha^{-1}(Im\,\gamma_i)$ with $s < t$ such that:

(1) the subpath $\alpha_{0:s}$ has seam list u;

(2) the subpath $\alpha_{t:1}$ has seam list v; and

(3) the subpath $\alpha_{s:t}$ crosses over no seam in Γ.

Write γ_i as γ, and define $a, b \in I$ by $\gamma(a) = \alpha(s)$ and $\gamma(b) = \alpha(t)$. Statement (3) implies that $\alpha_{s:t}$ lies within a single piece P of S. Since P is simply connected, and $\gamma_{a:b}$ is also a path in P, we have $\alpha_{s:t} \simeq_P \gamma_{a:b}$ by Lemma 2a.5. If γ is straight, then $\gamma_{a:b}$ is linear and hence $|\gamma_{a:b}| \leq |\alpha_{s:t}|$.

We create β by splicing $\gamma_{a:b}$ into α. Put $\beta(x) = \alpha(x)$ for $x \notin (s,t)$, and define $\beta_{s:t} = \gamma_{a:b}$. Clearly β is a PL path in S, and β is a link if α is. Also $\beta \simeq_P \alpha$ and—if γ is straight—$|\beta| \leq |\alpha|$, because of the corresponding facts about $\gamma_{a:b}$. Finally, since β does not cross over γ in the interval $[s,t]$, its seam list is that of $\alpha_{0:s}$ concatenated with that of $\alpha_{t:1}$. By statements (1) and (2) above, the seam list of β is just $u \cdot v$, which is ζ. □

If the seam list of α is ϵ, then β lies within in a single piece of S, and that piece is simply connected. Hence if α and β are loops, they are inessential.

Corollary 7a.3. *If a PL loop λ has empty path code in some pattern, then λ is inessential.* □

Construction of the blanket

The seams and pieces of a pattern form the arcs and nodes, respectively, of a graph: each seam is incident on the two pieces that include its image. If α is a PL chain that is free in the pattern, it corresponds to a path in this graph. The path begins with the unique piece containing $\alpha(0)$, ends with the unique piece containing $\alpha(1)$, and passes through a sequence of arcs equal to the seam list of α.

With this correspondence in mind, we show how to construct a blanket for a sheet S, given a pattern Γ for S. Pick any point $x_0 \in S$ that lies on none of the seams in Γ, and let Ξ denote the set of all path codes of PL paths beginning at x_0. By Lemma 7a.2, every path code is the seam list of some path, and we may take that path to be free. Hence every string $\xi \in \Xi$ corresponds to a path in the graph formed by the pieces of Γ. As such, it has a final piece *final*(ξ). Let P be the disjoint union of the final pieces of the seam lists in Ξ:

$$P = \bigoplus\nolimits_{\xi \in \Xi} \mathit{final}(\xi).$$

For each seam list ξ in Ξ, let h_ξ be a homeomorphism between the piece *final*(ξ) and the component of P corresponding to ξ. There is a natural projection $p\colon P \to S$ that sends $Im\, h_\xi$ to *final*(ξ) via h_ξ^{-1} for each $\xi \in \Xi$.

The blanket will be a quotient space Q of P. For each nonempty seam list ξ in Ξ, we have $\xi = \zeta \cdot \gamma_i$ for some i, and so *final*(ζ) and *final*(ξ) are adjacent pieces of S that share the thread $Im\,\gamma_i$. We identify $h_\zeta(\gamma_i(t))$ with $h_\xi(\gamma_i(t))$ for each point $t \in I$. If the point z lies on the seams $\gamma_1, \ldots, \gamma_k$, then $h_\xi(z)$ is identified with exactly k other points. Points lying on no seams are not identified with any others. Let Q be the quotient space resulting from these identifications, and let $q\colon P \to Q$ be the quotient map. By definition, a subset U is open in Q if and only if $q^{-1}(U)$ is open in P. The inverse image of a point of Q is mapped to a single point of S by p, and hence p factors through Q; say $p = s \circ q$.

$$\begin{tikzcd}
(P, h_\epsilon(x_0)) \arrow[rr, "q"] \arrow[dr, "p"'] & & (Q, q \circ h_\epsilon(x_0)) \arrow[dl, "s"] \\
& (S, x_0) &
\end{tikzcd} \tag{7–1}$$

We now prove that Q is a covering space of S. Later we show that Q is simply connected.

Claim 7a.4. *In diagram (7–1), the map $s\colon Q \to S$ is a covering map.*

Proof. Let z be any point of S. We must find a neighborhood U of z in S that is evenly covered by s. (See Definition 2b.1.) For historical reasons, we allow the possibility that two or more seams intersect at z. Assume without loss of generality that z lies on the seams $\gamma_1, \ldots, \gamma_m$, and that z lies in the pieces $S_0, \ldots, S_m$ for some $m \geq 0$. By suitably renumbering the seams and pieces, we may assume that $S_{i-1} \cap S_i = Im\,\gamma_i$ for $1 \leq i \leq m$. Let U be a neighborhood of z in S that intersects no seams other than $\gamma_1, \ldots, \gamma_m$ and no pieces other than $S_0, \ldots, S_m$.

We begin by characterizing $p^{-1}(U)$. Let Ξ_i be the subset of Ξ consisting of those path codes $\xi \in \Xi$ with *final*$(\xi) = S_i$. By the definition of p, we have

$$p^{-1}(U) = \bigcup_{i=0}^{m} \bigcup_{\xi \in \Xi_i} h_\xi(U \cap S_i). \tag{7–2}$$

The sets $h_\xi(U \cap S_i)$ are disjoint in P. We partition them into collections, each of which is sewn together by q to form a copy of U. For each $\xi \in \Xi_0$, and for $0 \leq i \leq m$, let ξ_i be the seam list obtained by reducing $\xi \cdot \gamma_1 \cdots \gamma_i$, removing consecutive occurrences of the same seam. Then ξ_i is the path code of some free PL path in S starting at x_0 and ending in S_i; hence we have $\xi_i \in \Xi_i$. For each seam list $\zeta \in \Xi_i$, there is a seam list $\xi \in \Xi_0$ such that $\xi_i = \zeta$: take ξ to be the reduction of $\zeta \cdot \gamma_i \cdots \gamma_1$. Hence the seam lists $\bigcup_{i=0}^{m} \Xi_i$ are in bijective correspondence with the seam lists $\{\, \xi_i : \xi \in \Xi_0 \text{ and } 0 \leq i \leq m \,\}$. Thus equation (7–2) can be put in the form

$$p^{-1}(U) = \bigcup_{\xi\in\Xi_0} \bigcup_{i=0}^{m} h_{\xi_i}(U\cap S_i). \tag{7-3}$$

Because q is onto, we have $s^{-1}(U) = q\circ q^{-1}\circ s^{-1}(U)$, which is $q(p^{-1}(U))$. For $\xi\in\Xi_0$, let V_ξ denote the open set $\bigcup_{i=0}^{m} h_{\xi_i}(U\cap S_i)$. The sets V_ξ are disjoint, and equation (7–3) implies that $s^{-1}(U) = \bigcup_{\xi\in\Xi_0} q(V_\xi)$.

To show that U is evenly covered by s, it suffices to show that the sets $q(V_\xi)$ are disjoint and homeomorphic to U under s. Hence we study the identifications that q makes within and between the collections V_ξ. Let ξ be a seam list in Ξ_0. For $1\le i\le m$ we have either $\xi_i = \xi_{i-1}\cdot\gamma_i$ (the usual case) or else $\xi_i\cdot\gamma_i = \xi_{i-1}$ (if the last i seams in ξ are $\gamma_i\cdots\gamma_1$). In either case, q identifies the set $h_{\xi_{i-1}}(U\cap Im\,\gamma_i)$ with the set $h_{\xi_i}(U\cap Im\,\gamma_i)$. That is,

$$q\circ h_{\xi_{i-1}}(z) = q\circ h_{\xi_i}(z), \qquad \text{for } z\in U\cap Im\,\gamma_i. \tag{7-4}$$

As a consequence, q identifies all the points $h_{\xi_i}(z)$ for $0\le i\le m$. It carries out no other identifications. Hence if ξ and ξ' are distinct elements of Ξ_0, then $q(V_\xi)$ and $q(V_{\xi'})$ are disjoint.

It remains to prove that for $\xi\in\Xi_0$, the map $s\colon q(V_\xi)\to U$ is a homeomorphism. We construct an inverse $r\colon U\to q(V_\xi)$ for s as follows. For $0\le i\le m$, define r on $U\cap S_i$ to be $q\circ h_{\xi_i}$. By equation (7–4), these definitions agree on their intersections. Since $U\cap S_i$ is closed in U, this makes r continuous. Now for $x\in U\cap S_i$ we have $s\circ r(x) = p\circ h_{\xi_i}(x) = x$, so $s\circ r = id_U$. Similarly, if $y\in q(V_\xi)$, then $y = q\circ h_{\xi_i}(x)$ for some i and some $x\in U\cap S_i$, by the definition of V_ξ. Then $r\circ s(y) = r(x) = y$, so $r\circ s$ is the identity on $q(V_\xi)$. □

Lifting to the blanket

The reason for constructing the blanket Q is to help us show that paths with different path codes have different lifts starting at the same point, and thus are not path-homotopic. The lifting is carried out by the following lemma.

Claim 7a.5. *Let ρ be a PL path in S with seam list ξ and path code ζ, and suppose $\rho(0)$ is the base point x_0. If Q is the covering space of S in diagram (7-1), there is a lift $\widetilde{\rho}$ of ρ to Q such that $\widetilde{\rho}(0) = q\circ h_\epsilon(\rho(0))$ and $\widetilde{\rho}(1) = q\circ h_\zeta(\rho(1))$.*

Proof. Put $\xi = \gamma_{i_1}\cdot\gamma_{i_2}\cdots\gamma_{i_k}$. For $0\le j\le k$, let ξ_j be the substring consisting of the first j seams of ξ, and let ζ_j be the reduction of that substring. Choose an ordered sequence of points $0 = t_0, t_1, \ldots, t_k, t_{k+1} = 1$ from I such that $\rho(t_j)\in Im\,\gamma_{i_j}$ for $1\le j\le k$, and each subpath $\rho_j = \rho_{t_j:t_{j+1}}$ lies within a piece of S, namely *final*(ζ_j). We lift ρ_j to the path

$$\widetilde{\rho}_j = q\circ h_{\zeta_j}\circ\rho_j,$$

which projects to ρ_j under s because $s \circ q \circ h_\theta$ is the identity on *final*(θ) for any reduced seam list θ. Then we define $\widetilde{\rho}$ by $\widetilde{\rho}_{t_j:t_{j+1}} = \widetilde{\rho}_j$ for $0 \leq j \leq k$. To show that $\widetilde{\rho}$ is well defined and continuous, we must prove that $\widetilde{\rho}_{j-1}(1) = \widetilde{\rho}_j(0)$ for $1 \leq j \leq k$. In other words, we must show that the points $h_{\zeta_{j-1}}(\rho_{j-1}(1))$ and $h_{\zeta_j}(\rho_j(0))$ are identified by q. They are, because $\rho_{j-1}(1) = \rho_j(0) \in Im\,\gamma_{i_j}$, and ζ_j differs from ζ_{j-1} only in a final seam γ_{i_j}.

The lemma now follows easily. The path $\widetilde{\rho}$ lifts ρ because $\widetilde{\rho}_j$ lifts ρ_j for each j. We also have $\widetilde{\rho}(0) = \widetilde{\rho}_0(0) = q \circ h_{\zeta_0}(\rho(0))$, and similarly $\widetilde{\rho}(1) = \widetilde{\rho}_k(1) = q \circ h_{\zeta_k}(\rho(1))$. Since $\zeta_0 = \epsilon$ and $\zeta_k = \zeta$, the endpoints of $\widetilde{\rho}$ are as desired. □

Now we can complete the proof that Q is a blanket of S.

Claim 7a.6. *In diagram (7–1), the space Q is simply connected.*

Proof. The space Q is path-connected because every piece of Q can be connected to the base point $h_\epsilon(x_0)$ by a path. Every point $z \in Q$ has the form $q \circ h_\zeta(y)$ where ζ is the reduced seam list of a PL path α from x_0 to a piece containing y. Let ν be a path in that piece from $\alpha(1)$ to y. Lifting α to Q via the preceding lemma, we obtain a path from $q \circ h_\epsilon(x_0)$ to $q \circ h_\zeta(\alpha(1))$, which, when concatenated with $q \circ h_\zeta \circ \nu$, connects $h_\epsilon(x_0)$ to z. Thus Q is path-connected.

To show that Q is simply connected, we prove that an arbitrary loop $\mu\colon I \to Q$ based at $q \circ h_\epsilon(x_0)$ is inessential. Because path homotopies can be lifted (Proposition 2b.4), it suffices to show that the loop $\lambda = s \circ \mu$ at x_0 *is inessential in* S. Any path in a sheet can be made piecewise linear by application of a path homotopy. (To prove it, cover the path with finitely many starlike regions.) Hence we may assume that λ is PL. Let ζ be the path code of λ. By the preceding lemma, λ has a lifting $\mu'\colon q \circ h_\epsilon(x_0) \rightsquigarrow q \circ h_\zeta(x_0)$. Uniqueness of liftings (Theorem 2b.2) tells us that $\mu = \mu'$. Hence $q \circ h_\zeta(x_0) \doteq q \circ h_\epsilon(x_0)$, and so ζ can differ from ϵ only by the seam (if any) that contains x_0. But ζ is reduced, and cannot begin or end with any such seam. Therefore $\zeta = \epsilon$. We conclude from Corollary 7a.3 that λ is inessential. □

Knowing that Q is a blanket, we can now extend Claim 7a.5 to say which seam liftings cut the lifting of a path ρ.

Lemma 7a.7. *Let S be a sheet with pattern Γ, and let ω be a PL path in S. There is a lifting $\widetilde{\omega}$ of ω such that the sequence of Γ-liftings that separate the the endpoints of $\widetilde{\omega}$, when projected to S, is the path code of ω.*

Proof. By Lemma 7a.3 there is a path $\rho \in [\omega]_P$ whose seam list and path code in the pattern Γ both equal the path code ζ of ω. To every lifting of ω there corresponds a lifting of $\widetilde{\rho}$ with the same endpoints. Hence it suffices to prove the lemma with ρ in place of ω.

We lift ρ to a blanket sewn together from pieces of Γ. Let S_0 be the first piece ρ enters. (If ρ stays entirely within a a seam, the lemma is trivial.) Let σ be a path in S_0 ending at $\rho(0)$ and beginning on no seam of Γ. The the seam list of $\sigma \star \rho$ is ζ. Choose the base point $\sigma(0)$ for S, and construct the blanket Q as in diagram (7-1). All blankets of S are equivalent, by Proposition 2b.7, so we may as well lift to Q. Lift $\sigma \star \rho$ to a path $\widetilde{\sigma} \star \widetilde{\rho}$ in Q as in Claim 7a.5. In the notation of Section 7A, we have $\widetilde{\rho}(0) = q \circ h_\epsilon(\rho(0))$. Say ζ is the sequence $\langle \gamma_1, \ldots, \gamma_n \rangle$, and for $1 \leq i \leq n$ let ζ_i be the subsequence $\langle \gamma_1, \ldots, \gamma_i \rangle$. Because ζ is reduced, ζ_i is a path code for each i. An examination of the specific lifting constructed by Claim 7a.5 tells us which seam liftings $\widetilde{\rho}$ crosses over. The ith seam lifting crossed over by $\widetilde{\rho}$ is $\widetilde{\gamma}_i = q \circ h_{\zeta_i} \circ \gamma_i$. Between $\widetilde{\gamma}_i$ and $\widetilde{\gamma}_{i+1}$ lies the lifting $Im\, q \circ h_{\zeta_i}$ of the piece $\mathit{final}(\zeta_i)$.

I argue that the links $\widetilde{\gamma}_1, \ldots, \widetilde{\gamma}_n$ are the seam liftings that separate the endpoints of $\widetilde{\rho}$, and that each link $\widetilde{\gamma}_i$ separates $\widetilde{\rho}(0)$ and the links $\widetilde{\gamma}_1, \ldots, \widetilde{\gamma}_{i-1}$ from $\widetilde{\rho}(1)$ and the links $\widetilde{\gamma}_{i+1}, \ldots, \widetilde{\gamma}_n$. All the links $\widetilde{\gamma}_i$ are distinct, which means $\widetilde{\rho}$ never crosses back over any of them. Thus each link $\widetilde{\gamma}_i$ separates the endpoints of $\widetilde{\rho}$. No other seam liftings do so, else $\widetilde{\rho}$ would cross over them. Finally, for $1 < i < n$ the piece lifting shared by $\widetilde{\gamma}_{i-1}$ and $\widetilde{\gamma}_i$ lies on the opposite side of $\widetilde{\gamma}_i$ from the piece lifting shared by $\widetilde{\gamma}_i$ and $\widetilde{\gamma}_{i+1}$. Hence $\widetilde{\gamma}_i$ separates the sets $Im\, \widetilde{\gamma}_j$ with $j > i$ from the sets $Im\, \widetilde{\gamma}_j$ with $j < i$. □

Path homotopy and link homotopy

Results 7a.2 through 7a.5 imply that path codes characterize path homotopy, at least for free paths.

Proposition 7a.8. *Let Γ be a pattern for a sheet S. Two PL paths in S that are free in Γ are path-homotopic if and only if they have the same endpoints and the same path code in Γ.*

Proof. Let α and α' be piecewise linear paths in S. If α and α' have different endpoints, then they cannot be path-homotopic, so we assume henceforth that $\alpha(0) = \alpha'(0)$ and $\alpha(1) = \alpha'(1)$. Let ξ and ξ' be the path codes of α and α', respectively, in the pattern Γ. By Lemma 7a.2, there are PL paths $\beta \in [\alpha]_P$ and $\beta' \in [\alpha']_P$ whose seam lists are ξ and ξ', respectively. It suffices to prove that $\beta \simeq_P \beta'$ if and only if $\xi = \xi'$.

First we prove the "if" direction. Suppose $\xi = \xi'$, and let λ be the loop $\widehat{\beta} \star \beta'$. Because β and β' are free in Γ, the loop λ does not cross over any seam at 1/2. Hence the seam list of λ is then that of $\widehat{\beta}$, namely $\widehat{\xi}$, concatenated with that of β', namely ξ', which equals ξ. It follows that the path code of λ is empty, which makes λ inessential (Corollary 7a.3). By the groupoid properties of concatenation (Section 2A), we have

$$[\beta]_P = [\beta \star \lambda]_P = [\beta \star (\widehat{\beta} \star \beta')]_P = [\beta']_P.$$

Now we prove the "only if" direction. Let x_0 be the point $\beta(0)$ of S, and let Q be the covering space of S constructed by the sewing technique leading to diagram (1). Suppose β and β' are path-homotopic, and let z denote the point $\beta(1) = \beta'(1)$. We now lift β and β' to Q beginning at the point $q \circ h_\epsilon(x_0)$. Let $\widetilde{\beta}$ and $\widetilde{\beta}'$ be the lifts of β and β' given by Claim 7a.5. Then $\widetilde{\beta}(1) = q \circ h_\xi(z)$ and $\widetilde{\beta}'(1) = q \circ h_{\xi'}(z)$. Because $\beta \simeq_P \beta'$, we have $\widetilde{\beta} \simeq_P \widetilde{\beta}'$ by Proposition 2b.4, and hence $q \circ h_\xi(z)$ and $q \circ h_{\xi'}(z)$ are the same point of Q. In other words, $h_\xi(z)$ and $h_{\xi'}(z)$ are identified by q. Since β and β' are free, z lies on no seam, and therefore $\xi = \xi'$. □

Characterizing link homotopy in terms of seam lists is more difficult, but one useful result is relatively easy. Let Γ be a pattern for the sheet S. The **borders** of a piece P of Γ are the components of $P \cap Bd\, S$. If a link β in S is free in Γ, then each of its endpoints lies in exactly one border, and these borders are called the **roots** of β. Like the endpoints of β and the terminals of β, we consider the roots of β to be an ordered pair.

Lemma 7a.9. *Let Γ be a pattern of disjoint seams for the sheet S. If two free links in S have the same path code and the same roots in Γ, they are link-homotopic.*

Proof. This claim follows directly from Lemma 3a.4 and Proposition 7a.8. If two free links α and β have the same borders P and Q, there are paths $\nu\colon \alpha(0) \rightsquigarrow \beta(0)$ and $\kappa\colon \alpha(1) \rightarrow \beta(1)$ in P and Q, respectively, which touch no seams. And if α and β have the same seam list in Γ, then $\alpha \star \kappa$ and $\nu \star \beta$, which have the same endpoints, have the same seam list in Γ as well. Proposition 7a.8 now shows $\alpha \star \kappa \simeq_P \nu \star \beta$, which by Lemma 3a.4 implies $\alpha \simeq_P \beta$. □

7B. Crossing Sequences

In the last section we related the homotopy classes of paths and links to sequences of cuts. Previous results have shown us, however, that knowing the identity of a cut is often insufficient; one must also have a lifting or a crossing of that cut. Consequently, this section studies crossing sequences of paths and relates them to path codes. Two types of crossing sequences are of special interest: crossing sequences of cuts in designs, and crossing sequences of wires in patterns. We handle both in a single framework by studying **arrangements**, which generalize both designs and patterns. An arrangement on a sheet S is a finite set of disjoint simple cuts in S.

This section presents two main results that stand on their own, and several smaller results whose importance will be clearer in the next section. One result is a formal definition of the *content* of a cut, which in the design model is the sequence of wires forced to cross it. We show that in all embeddings of a design that minimize the number of crossings of a simple cut χ, the sequence of wires crossing χ is the content of χ (up to link homotopy). Another theorem says that for any arrangement of disjoint cuts of a design, there is an embedding of that design (not necessarily proper) in which each wire makes as few crossings with those cuts as possible. The latter result is the basis for a procedure used within the sketch compaction algorithm for computing flow.

Lists, codes, and plans

Before we plunge into the definitions, a word about terminology is in order. Given a chain α for a link in a sheet, and given a collection Φ of chains for links in the same sheet, there are several sequences one can define. If the elements of Φ are disjoint, one is the sequence of paths in Φ that α crosses over. We call this the *seam list*, *cut list*, or *wire list* of α in Φ, according to whether the elements of Φ are thought of as seams, cuts, or wires. We also define *codes* of α in Φ, which are always subsequences of the list of α in Φ. The *path code* (or *link code*) of α in Φ is the sequence of elements of Φ that α is forced to cross, thinking of its endpoints (or its terminals) as fixed. The final type of sequence we consider is a sequence of crossings. Because the paths in the collection Φ will sometimes intersect, the definition of crossing sequence is not the obvious one. A **crossing sequence** or **plan** for α in Φ is a finite sequence of triples (ρ, a, t) such that $\rho \in \Phi$ and (a, t) is a crossing of ρ by α. If the crossings of α by cuts in Γ are finite in number, and no two occur at the same point of ω, then we may speak of the **full** plan of ω in Γ, namely the set of crossings of ω by cuts in Γ, ordered by position along ω. But we are primarily concerned with the kinds of crossing sequences in the two-part definition below.

Definition 7b.1. Let M be the blanket of a sheet S, with $p: M \rightarrow S$ the covering map. Let ω be a chain for a link in S. Let $\tilde{\omega}$ be a lifting of ω to M, and let Γ be a arrangement on S. Consider the Γ-liftings that separate either (a) the endpoints of ω, or (b) the terminals of $\tilde{\omega}$. They have a unique ordering $\tilde{\gamma}_1, \ldots, \tilde{\gamma}_n$ such that for $1 \leq k \leq n$, the simple link $\tilde{\gamma}_k$ separates $\tilde{\omega}(0)$ and $\tilde{\gamma}_1$ through $\tilde{\gamma}_{k-1}$ from $\tilde{\omega}(1)$ and $\tilde{\gamma}_{k+1}$ through $\tilde{\gamma}_n$. We call the sequence $\langle \gamma_1, \ldots, \gamma_n \rangle$ the (a) **path code** or (b) **link code** of ω in Γ. For $1 \leq k \leq n$, let (c_k, t_k) be a crossing of $\tilde{\gamma}_k$ by $\tilde{\omega}$. The sequence of length n whose kth element is $(p \circ \tilde{\gamma}_k, c_k, t_k)$ is a (a) **path plan** or (b) **link plan** for ω in Γ.

When Γ is a pattern, this definition of path code agrees with that given in Section 7A, by Lemma 7a.7.

Kinship of crossing sequences

Path plans and link plans are not unique in general, but there is one important situation in which a link has only one link plan. We say a link ω **conforms with** an arrangement Γ if $cross(\gamma,\omega) = wind(\gamma,\omega)$ for all cuts $\gamma \in \Gamma$. Said another way, ω conforms with Γ if and only if all crossings of cuts in Γ by ω are necessary and no two are similar. In this case the unique link plan of ω in Γ is the full plan of ω in Γ.

When a path has more than one path plan or link plan, all those plans are related by *kinship*. Let γ, γ', ω, and ω' be chains for links in a sheet S. (In other words, they all begin and end on $Bd\,S$, and so each has two terminals.) A crossing (c,t) of γ by ω is **akin** to a crossing (c',t') of γ' by ω' if whenever $\widetilde{\gamma}$ and $\widetilde{\omega}$ are liftings of γ and ω that reflect (c,t), there there are liftings $\widetilde{\gamma}'$ and $\widetilde{\omega}'$ reflecting (c,t) such that $\widetilde{\gamma}$ has the same terminals as $\widetilde{\gamma}'$ and $\widetilde{\omega}$ has the same terminals as $\widetilde{\omega}'$. Two plans are **akin** if they have the same length, say n, and for $1 \leq k \leq n$, the kth crossing in one is akin to the kth crossing of the other.

Lemma 7b.2. *Let ρ be a chain for a link ω, let an arrangement Γ be given. All the paths in $[\rho]_P$ have the same path code in Γ, and all their path plans in Γ are akin. All the chains for links in $[\omega]_L$ have the same link code in Γ, and all their link plans in Γ are akin.*

Proof. These claims follow directly from Definition 7b.1. If $\alpha \simeq_P \beta$, one can choose liftings $\widetilde{\alpha}$ and $\widetilde{\beta}$ that have the same endpoints. Hence the sequence $\langle \widetilde{\gamma}_1, \ldots, \widetilde{\gamma}_n \rangle$ in Definition 7b.1 will be the same for both, and corresponding crossings will be akin. Similarly, if α and β are chains such that $\alpha \simeq_P \alpha' \simeq_L \beta' \simeq_P \beta$ for some links α' and β', then one can choose liftings $\widetilde{\alpha}$ and $\widetilde{\beta}$ that have the same terminals. Consequently the sequence $\langle \widetilde{\gamma}_1, \ldots, \widetilde{\gamma}_n \rangle$ will be the same for both, and again, corresponding crossings will be akin. □

Kinship of crossings is strongly related to kinship of subcuts. Suppose that ω and ω' are routes for wires, that γ and γ' are cuts, and that the crossing (c,t) of γ by ω is akin to the crossing (c',t') of γ' by ω'. Then for $e \in \{0,1\}$, the half-cut $\alpha_{e:c}$ for ω at t is akin to the half-cut $\alpha'_{e:c'}$ for ω' at t'.

Link plans versus path plans

Every link plan for a link is the result of removing some of the crossings in a path plan for that link. Let ω and γ be paths in a sheet S that begin and end on $Bd\,S$. (They are chains for links in S.) A crossing (c,t) of γ by ω is **trivial** if for some $i,j \in \{0,1\}$ the path $\gamma_{i:c} \star \omega_{t:j}$ is trivial—path-homotopic to a path in $Bd\,S$. We call the crossing **0-trivial** if $j = 0$ and **1-trivial** if $j = 1$. Link plans are obtained from path plans by deleting as many initial 0-trivial crossings and as many final 1-trivial crossings as possible.

Lemma 7b.3. *Let ω be a chain for a link. Every path plan for ω consists of a sequence of* 0*-trivial crossings, followed by a sequence of nontrivial crossings, followed by a sequence of* 1*-trivial crossings. Every link plan for ω consists of the nontrivial crossings in some path plan of ω.*

Proof. We adopt the notation of case (a) of Definition 7b.1, so $\langle \tilde{\gamma}_1, \ldots, \tilde{\gamma}_n \rangle$ is the sequence of Γ-liftings that separate the endpoints of the lift $\tilde{\omega}$. Link plans are defined in terms of the subsequence of $\langle \tilde{\gamma}_1, \ldots, \tilde{\gamma}_n \rangle$ consisting of those cut liftings that separate the *terminals* of $\tilde{\omega}$. Call these terminals T_0 and T_1, where $\tilde{\omega}(j) \in T_j$ for $j \in \{0,1\}$. A link $\tilde{\gamma}_k$ fails to separate T_0 from T_1 if and only if $\tilde{\gamma}_k$ shares a terminal with $\tilde{\omega}$, meaning that $\tilde{\gamma}_k(i)$ lies on the same fringe T_j as $\tilde{\omega}(j)$ for some $i, j \in \{0,1\}$. And the latter is true if and only if (c_k, t_k) is a j-trivial crossing of γ_k by ω. Therefore a typical link plan for ω in Γ is obtained from a path plar for ω in Γ by removing its trivial crossings and nothing more.

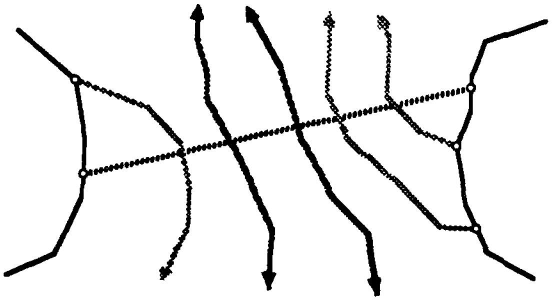


Figure 7b-1. *The link plan within the path plan.* The light grey and dark grey links are the Γ-liftings that separate the endpoints of $\tilde{\omega}$; the darker ones also separate the terminals of $\tilde{\omega}$. The earlier light grey links make 0-trivial crossings with $\tilde{\omega}$, and the later ones make 1-trivial crossings.

Now we argue that the link plan is a contiguous subsequence of the path plan. Let $1 \leq i < j < k \leq n$, and suppose both $\tilde{\gamma}_i$ and $\tilde{\gamma}_k$ separate the terminals of $\tilde{\omega}$. It suffices to show that $\tilde{\gamma}_j$ does also. According to Definition 7b.1, the link $\tilde{\gamma}_i$ separates the fringe containing $\tilde{\omega}(0)$ from $\tilde{\gamma}_j$, and likewise $\tilde{\gamma}_k$ separates the fringe containing $\tilde{\omega}(1)$ from $\tilde{\gamma}_j$. Since $\tilde{\omega}(0)$ and $\tilde{\omega}(1)$ lie on opposite sides of $\tilde{\gamma}_j$, it follows that the entire scrap of $\tilde{\gamma}_i$ containing $\tilde{\omega}(0)$ is on the opposite side of $\tilde{\gamma}_j$ from the scrap of $\tilde{\gamma}_k$ containing $\tilde{\omega}(1)$. In particular, the terminals of $\tilde{\omega}$ lie wholly on opposite sides of $\tilde{\gamma}_j$.

It remains only to show that no 0-trivial crossing in the path plan for ω follows a crossing that is not 0-trivial. Suppose that the ith crossing is not 0-trivial, which means that $\tilde{\gamma}_i$ does not intersect T_0. For $j > i$, the link $\tilde{\gamma}_i$ separates $\tilde{\gamma}_j$ from $\tilde{\omega}(0)$ and hence from T_0. Consequently $\tilde{\gamma}_j$ cannot be 0-trivial, and the proof is complete. □

The content of a cut

We now take the arrangement to be a design. Let χ be a cut of a design Ω. The link code of χ in Ω is what I call the **content** of χ in Ω. Lemma 7b.2 says that link-homotopic links have equal content. Content determines flow: If the content of χ in Ω is the sequence $\langle \omega_1, \ldots, \omega_n \rangle$, then we have

$$\mathit{flow}(\chi, \Omega) = \sum_{i=1}^{n} \mathit{width}(\omega_i).$$

The link plans of χ in Ω are even more useful than content, however, because they determine the flows across half-cuts also.

Lemma 7b.4. *Let χ be a cut of a design Ω, and suppose the link plan of χ in Ω includes n crossings. If for $1 \leq k \leq n$ the kth crossing is (ω_k, t_k, a_k), then*

$$\mathit{flow}(\chi_{0:a_k}, \Omega) = \sum_{i=1}^{k-1} \mathit{width}(\omega_i) \qquad \text{and} \qquad \mathit{flow}(\chi_{1:a_k}, \Omega) = \sum_{i=k+1}^{n} \mathit{width}(\omega_i).$$

Proof. The proof is a direct computation from the definition of flow. Let $\tilde{\chi}$ be a lift of χ, and for $1 \leq k \leq n$ lift ω_k to $\tilde{\omega}_k$ so that $\tilde{\chi}(a_k) = \tilde{\omega}_k(t_k)$. By the definition of link plan, the links $\tilde{\omega}_k$ are precisely the Ω-liftings that cut $\tilde{\chi}$. Consequently the flow across χ is just $\sum_{i=1}^{n} \mathit{width}(\omega_i)$. The method of Proposition 4d.2 shows that the liftings that contribute to the flow across $\chi_{0:a_k}$ are just $\tilde{\omega}_1$ through $\tilde{\omega}_{k-1}$, and the liftings that contribute to the flow across $\chi_{1:a_k}$ are just $\tilde{\omega}_{k+1}$ through $\tilde{\omega}_n$. □

When one replaces a design by an embedding of a design, the only effect on the content of a cut is to replace each wire by its new embedding. In fact, something stronger is true: the link plans of a cut in different embeddings of a design are all akin.

Lemma 7b.5. *Let Υ be an embedding of a design Ω, and let χ be a cut of these designs. The link plan of χ in Ω is akin to that of χ in Υ.*

Proof. Let S be the sheet and M its blanket. Lift χ to a simple link $\tilde{\chi}$ in M. Write $n = \sum_{\omega \in \Omega} \mathit{wind}(\gamma, \omega) = \sum_{v \in \Upsilon} \mathit{wind}(\gamma, v)$. Let $\tilde{\omega}_1, \ldots, \tilde{\omega}_n$ be the Ω-liftings that cut $\tilde{\chi}$, ordered as in Definition 7b.1, and likewise let $\tilde{v}_1, \ldots, \tilde{v}_n$ be the Υ-liftings that cut $\tilde{\chi}$. No two of the links $\tilde{\omega}_1, \ldots, \tilde{\omega}_n$ are link-homotopic, by Lemma 4c.3, and similarly for $\tilde{v}_1, \ldots, \tilde{v}_n$. Because each wire in Υ is link-homotopic to a wire in Ω, Proposition 3a.6 implies that there is a permutation π of $\{1, \ldots, n\}$ such that $\tilde{\omega}_i \simeq_P \tilde{v}_{\pi(i)}$ for each i.

We prove that π is the identity permutation. Since the wires in Ω are simple and disjoint, the links $\tilde{\omega}_1, \ldots, \tilde{\omega}_n$ are disjoint. Each makes exactly one crossing with $\tilde{\gamma}$, and it crosses over $\tilde{\gamma}$ there, so for each i the links $\tilde{\omega}_j$ with $j < i$ lie in on the opposite side of $\tilde{\omega}_i$ from the links $\tilde{\omega}_j$ with $j > i$. The cut γ respects Ω weakly because γ is simple (Proposition 4c.7), and therefore no two liftings $\tilde{\omega}_i$ and $\tilde{\omega}_j$ share a terminal. Hence for each i, the terminals of the links $\tilde{\omega}_j$ with $j < i$ lie in the opposite scrap of

$\widetilde{\omega}_i$ from the terminals of the links $\widetilde{\omega}_j$ with $j > i$. A symmetrical statement holds for $\widetilde{v}_i$. Let Ξ denote the collection of terminals of the links $\widetilde{\omega}_1, \ldots, \widetilde{\omega}_n$, which are also the terminals of the links $\widetilde{v}_1, \ldots, \widetilde{v}_n$. The number of fringes in Ξ which lie in the scrap of $\widetilde{\omega}_i$ that contains $\widetilde{\gamma}(0)$ is precisely $2(i-1)$. The same goes for $\widetilde{v}_i$. But by Proposition 3c.4, the links $\widetilde{\omega}_i$ and $\widetilde{v}_{\pi(i)}$, being link-homotopic, separate the fringes equally. We conclude that $\pi(i) = i$ for each i. The lemma follows. □

Designs that minimize crossings

Next we study embeddings of designs in which the wires cross certain simple cuts as seldom as possible. Let Ω be a design on the sheet S, and let Γ be an arrangement of cuts in S. The design Ω is **stable** with respect to Γ if wherever a wire of Ω crosses a cut in Γ, it crosses *over* the cut there. (Consequently $cross(\gamma, \omega)$ is finite for all $\gamma \in \Gamma$ and all $\omega \in \Omega$.) We say that Ω **conforms with** Γ if for every wire $\omega \in \Omega$ and every cut $\gamma \in \Gamma$ we have $cross(\gamma, \omega) = wind(\gamma, \omega)$. In other words, every crossing of γ by ω must be necessary, and no two may be similar. Conformity implies stability: If Ω conforms with Γ, then Ω is also stable with respect to Γ.

If the design Ω does not conform with Γ, then some wire in Ω *deviates* across some cut in Γ. Pick $\omega \in \Omega$ and $\gamma \in \Gamma$. Let $\omega_{s:t}$ and $\gamma_{a:b}$ be subpaths such that either (1) $\omega_{s:t} \simeq_P \gamma_{a:b}$ or (2) $\omega_{t:s} \star \gamma_{a:b}$ is a trivial link. Then we call $\omega_{s:t}$ a **deviation** across $\gamma_{a:b}$.

Lemma 7b.6. *If a link ω does not conform with a cut γ, then some subpath of α is a deviation across some subpath of γ.*

Proof. Suppose $cross(\gamma, \omega) \neq wind(\gamma, \omega)$. By the definition of winding, the quantity $cross(\gamma, \omega)$ is the larger. Either ω makes an unnecessary crossing with γ, or else two crossings of γ by ω are similar. If possible, choose two similar crossings (a, s) and (b, t) of γ by ω. In this case $\omega_{s:t} \simeq_P \gamma_{a:b}$. Otherwise let (a, s) be an unnecessary crossing of γ by ω, and let $\widetilde{\gamma}$ and $\widetilde{\omega}$ be liftings that reflect this crossing. Because no crossing of γ by ω is similar to (a, s), the link $\widetilde{\omega}$ makes only the one crossing (a, s) with $\widetilde{\gamma}$. Hence $\widetilde{\omega}$ shares a terminal with $\widetilde{\gamma}$; say $\widetilde{\omega}(t)$ lies on the same fringe as $\widetilde{\gamma}(b)$. In this case $\omega_{t:s} \star \gamma_{a:b}$ is a trivial link. So in either case $\omega_{s:t}$ is a deviation across $\gamma_{a:b}$. □

Making a design conform to an arrangement

Given a design Ω and an arrangement Γ on the same sheet, we prove that Ω has an embedding Υ that conforms with Γ. It can be obtained from Ω by a sequence of local alterations that remove *collapsible* subpaths of wires in Ω. Suppose $\omega_{s:t}$ is a deviation across $\gamma_{a:b}$. If $\omega_{s:t}$ is clean in Γ, meaning that $Mid\, \omega_{s:t}$ intersects no cut in Γ, and $\gamma_{a:b}$ is clean in Ω, then we say $\omega_{s:t}$ is **collapsible** to $\gamma_{a:b}$ and vice versa. If

$\omega_{s:t}$ is collapsible, then at least one crossing of γ by ω can be removed by routing ω, and after this routing Ω is still a design. One simply splices the path $\gamma_{a:b}$ into ω in place of $\omega_{s:t}$, and then displaces this subpath slightly away from γ. If Ω was previously stable with respect to Γ, it still is.

To make Ω conform with Υ, one first finds an embedding Υ of Ω that is stable with respect to Γ. That step is easy. One then repeatedly collapses subpaths of wires in Ω until no more collapsible subpaths exist. The crossings between wires of Ω and cuts in Γ are finite in number, and at least one is removed with each collapsible subpath. Hence the collapsing process must terminate. The following lemma says that it terminates in a design that conforms with Γ.

Proposition 7b.7. *Suppose the design Ω is stable with respect to an arrangement Γ. If no wire in Ω has a collapsible subpath, then Ω conforms with Γ.*

Proof. We prove the contrapositive. Supposing that Ω does not conform with Γ, we find a collapsible subpath of a wire in Ω. Suppose that the wire $\omega \in \Omega$ does not conform with the cut $\gamma \in \Gamma$. By Lemma 7b.6, there is a deviation $\omega_{s:t}$ across a subpath $\gamma_{a:b}$.

First we make $\gamma_{a:b}$ clean in Ω. Suppose it is not. Let $\tilde{\gamma}$ and $\tilde{\omega}$ be lifts of γ and ω such that $\tilde{\gamma}(a) = \tilde{\omega}(s)$. If $\tilde{\omega}$ makes another crossing (b, t) with $\tilde{\gamma}$, we may choose it so that $\tilde{\omega}$ does not touch the middle of $\tilde{\gamma}_{a:b}$. Let λ be the path $\omega_{s:t} \star \gamma_{b:a}$. Then either λ is a simple loop (if (b, t) is a crossing) or else $Im\,\lambda$ is a web of one thread. In either case λ has an inside N that contains no fringes (Propositions 3b.8 and 3c.5). Because $\gamma_{a:b}$ is not clean in Ω, there is a wire $v \in \Omega$ and a crossing (c, x) of γ by v where $c \in (a, b)$. Lift v to $\tilde{v}$ so that $\tilde{\gamma}(c) = \tilde{v}(x)$. Because Ω is stable with respect to Γ, the link $\tilde{v}$ crosses over $\tilde{\gamma}$ at x, and consequently it enters N. Choose a point y so that $Mid\,\tilde{v}_{x:y} \subset N$. The point $\tilde{v}(y)$ must fall on $\tilde{\gamma}_{a:b}$, or on a fringe shared by $\tilde{\gamma}$ and $\tilde{\omega}$. Put $d = \tilde{\gamma}^{-1}(\tilde{v}(y))$ or $d = b$ accordingly. Then $v_{x:y}$ is a deviation across $\gamma_{c:d}$. Moreover, the interval $[c, d]$ is strictly contained within $[a, b]$, and the number of such intervals (delimited by crossings with wires in Ω) is finite. So if we replace $\omega_{s:t}$ and $\gamma_{a:b}$ by $v_{x:y}$ and $\gamma_{c:d}$, and repeat, we eventually get stuck with a clean subpath of γ.

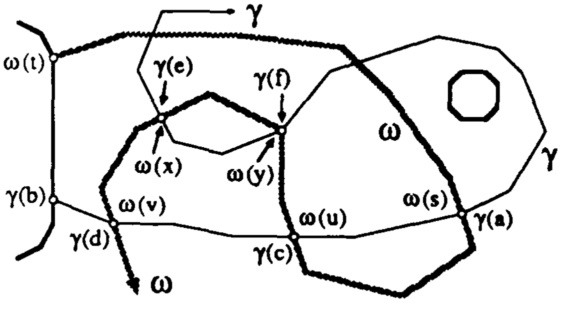


Figure 7b-2. *Finding the collapsible subpath.* Grey paths are wires, striped paths are cuts. Starting with the deviation $\omega_{s:t}$ across $\gamma_{a:b}$, we move to the deviation $\omega_{u:v}$ across $\gamma_{c:d}$, and thence to the collapsible deviation $\omega_{x:y}$ across $\gamma_{e:f}$.

Now we use the same ideas to make $\omega_{s:t}$ clean in Γ. Suppose the cut $\chi \in \Gamma$ makes a crossing (c, x) with ω, where $x \in (s, t)$. Define the liftings $\tilde{\gamma}$ and $\tilde{\omega}$ and the submanifold N as before, and let $\tilde{\chi}$ be a lift of χ such that $\tilde{\chi}(c) = \tilde{\omega}(x)$. Then $\tilde{\chi}$ enters N at x, and since the cuts in Γ are disjoint, it must leave at a crossing (d, y) with $\tilde{\omega}$ or end at a point $\tilde{\chi}(d)$ on the fringe containing $\tilde{\omega}(x)$, where $x = t$. In either case $\omega_{x:y}$ is a deviation across $\chi_{c:d}$. Because $\gamma_{a:b}$ is clean in Ω, no wires of Ω can penetrate N, and so $\chi_{c:d}$ is clean in Ω. If we replace $\omega_{s:t}$ and $\gamma_{a:b}$ by $\omega_{x:y}$ and $\chi_{c:d}$, and repeat, we end up with a clean deviation of ω across a clean subpath of γ. This deviation is collapsible. □

Corollary 7b.8. *If Γ is an arrangement on the sheet of a design Ω, then some embedding of Ω conforms with Γ.* □

This result extends Lemma 4b.5, which says that if ω is a wire and χ is a cut, then ω has a route that crosses χ at most $\mathit{wind}(\chi, \omega)$ times. Corollary 7b.8 is stronger in most respects, for it ensures that the route is an embedding, and it handles many wires and many cuts simultaneously. Its only drawback is that the path ω must be a wire, not just any link.

In combination with Lemma 7b.5, Corollary 7b.8 provides an alternate definition of content. If a design Υ conforms with a cut γ, then the full plan of γ in Υ is also its link plan, and hence the content of γ in Υ is just the sequence of wires in Υ that γ crosses. The content of γ in a design Ω therefore is the unique sequence of wires $\omega_1, \ldots, \omega_n$ taken from Ω such that for every embedding Υ of Ω that conforms with Γ, the sequence of wires that γ crosses in Υ is link-homotopic, element by element, to $\langle \omega_1, \ldots, \omega_n \rangle$.

7C. Two Methods for Computing Plans

Building on the results of the preceding section, we now present two methods for computing a link plan for a cut with respect to a design and a path plan for a wire with respect to a pattern. (Actually, we compute plans akin to these.) The former type of plan tells us the content of a cut, from which we can compute its flow and the flows of certain half-cuts. The latter type of plan helps us determine the corridors through which to route wires. Both methods solve both problems. One of the two methods involves replacing wires by their elastic chains, and it helps justify the use of the rubber-band equivalent in Chapter 1. The other method involves finding an embedding of one's design that conforms with a certain pattern. We will use it in Chapter 9 when we study compaction.

The elastic-chain equivalent

By analogy with the rubber-band equivalent of a sketch, I define the **elastic-chain equivalents** of a design. Every path in a sheet, and every wire in particular, has a unique elastic chain. Let us denote the elastic chain of a wire ω by the symbol $\overline{\omega}$. An elastic-chain equivalent of a design Ω is obtained by replacing each wire $\omega \in \Omega$ by the elastic chain ρ for a link in $[\omega]_L$. Later I describe how to add information to this structure to support computation of plans.

Elastic-chain equivalents have three important properties that we discuss here. First, one can compute the elastic chain ρ by choosing a pattern of convex pieces and finding a minimum-length path from $\omega(0)$ to $\omega(1)$ whose seam list is the path code of ω. Second, the elastic chain ρ has an easily computable path plan in any arrangement of straight cuts, which is just its cut list in that arrangement. The third property concerns a cut χ and an arbitrary elastic-chain equivalent Φ of a design Ω. I show how to sort the nontrivial crossings of χ by the elastic chains in Φ to form a crossing sequence akin to the link plans for χ in Ω.

Constructing elastic chains

The results of Section 7A give us a means of characterizing elastic chains.

Lemma 7c.1. *Let Γ be a pattern for a sheet S, and let ω be a wire in S that is free in Γ. The elastic chain for ω is the shortest canonical path in S from $\omega(0)$ to $\omega(1)$ whose seam list in Γ is the path code of ω in Γ.*

Proof. Denote by ζ the path code of ω in Γ, and let ρ be the elastic chain for ω. By Proposition 7a.8, every chain for ω is a path from $\omega(0)$ to $\omega(1)$ whose path code in Γ is ζ. By the definition of elastic chain, ρ is the shortest such path that is canonical. Lemma 7a.2 shows that every path with path code ζ is path-homotopic to a path with seam list ζ that is no longer. Hence the seam list of ρ is ζ. □

Lemma 7c.1 suggests a method of constructing elastic chains. One first finds a pattern Γ for the sheet S whose pieces are convex regions. (Its seams must be straight.) To find the elastic chain for a path α, one then constructs the path code ζ of α with respect to Γ by computing its seam list and performing the appropriate reductions. The seams in the path code of α—or rather, their images—form a corridor. The elastic chain for α can be computed by Algorithm W: it is the shortest canonical path β through this corridor from $\alpha(0)$ to $\alpha(1)$. (See Section 1B.) How do we know this? Because the pieces of Γ are convex and β has minimum length, β crosses over no fringe edges and no seams except those in ζ. Hence β is a path in S with seam list ζ. Conversely, any path in S with seam list ζ is a path through the corridor defining β. Thus β is the shortest canonical path from $\alpha(0)$ to $\alpha(1)$ whose seam list is ζ.

Path plans of elastic chains

Intuitively, the main reason for looking at the elastic-chain equivalent is that no elastic chain crosses any straight cut any more often than necessary. We provide two powerful formulations of this statement: one in terms of the path plans of an elastic chain with respect to straight cuts, and one in terms of the link plans of straight cuts with respect to elastic chains. Both rest upon the following basic lemma.

Lemma 7c.2. *If α and β lift elastic chains for links, then β crosses over α only if the endpoints of β lie on opposite sides of α.*

Proof. We first derive an easy fact about pairs of crossings of γ by ρ. Let p denote the covering map, and let (a, s) and (b, t) be two crossings of α by β. Lemma 2a.5 we have $\alpha_{a:b} \simeq_P \beta_{s:t}$. Both $p \circ \alpha$ and $p \circ \beta$ are elastic, and so by Lemma 3d.2, the paths $(p \circ \alpha)_{a:b}$ and $(p \circ \beta)_{t:t'}$ are both elastic. And since they are path-homotopic, Lemma 3d.7 shows that they are identical. Hence $\alpha_{a:b} = \beta_{s:t}$ by uniqueness of liftings.

One conclusion is that β crosses over α at most once. For if it crossed over twice, there would be crossings (a, s) and (b, t) of α by β such that $Im\,\beta_{s:t} \not\subseteq Im\,\alpha$, and this possibility we just ruled out. Suppose now that the endpoints of β do not lie on opposite sides of α, meaning that for every link $\chi \in [\alpha]_P$, the points $\beta(0)$ and $\beta(1)$ do not lie on opposite sides of χ. If β were to cross over α, once, then we could find a simple link $\chi \in [\alpha]_P$ over which β crossed exactly once, and hence the endpoints of β would lie on opposite sides of χ. □

Because straight cuts are elastic, Lemma 7c.2 implies that a lifting of an elastic chain crosses each straight link at most once. Hence if ρ is an elastic chain and Γ is an arrangement of straight cuts, the *cut list* of ρ in Γ—the sequence of cuts in Γ that it crosses over—is also its path code in Γ. If we choose one crossing from each interval in which ρ crosses over a cut of Γ, we get a **wire plan** for ρ in Γ. Wire plans are nothing more than path plans.

Lemma 7c.3. *Let ρ be an elastic chain in a sheet S, and let Γ be a disjoint arrangement of straight cuts in S. Every wire plan of ρ in Γ is a path plan for ρ in Γ.*

Proof. Let $\widetilde{\rho}$ be any lifting of ρ, and let $\widetilde{\gamma}$ lift a cut in Γ. Straight paths are elastic, and hence Lemma 7c.2 applies to $\widetilde{\rho}$ and $\widetilde{\gamma}$. It says that $\widetilde{\rho}$ crosses over $\widetilde{\gamma}$ only if $\widetilde{\gamma}$ separates the endpoints of $\widetilde{\rho}$, and then only once. (Of course, $\widetilde{\rho}$ does cross over $\widetilde{\gamma}$ if $\widetilde{\gamma}$ separates its endpoints.) Furthermore, $\widetilde{\rho}$ crosses over $\widetilde{\gamma}$ at a point t if and only if ρ crosses over γ at t. Hence for every wire plan of ρ, the crossings in that sequence are the same as the crossings in some path plan for ρ. Since that path plan is akin to all the others, by Lemma 7b.2, it remains to show that the crossings are ordered identically in the two lists. Let $\widetilde{\gamma}_1, \ldots, \widetilde{\gamma}_n$ be the Γ-liftings that separate the endpoints of ρ, ordered as in Definition 7b.1. These simple links are are disjoint, and each one separates $\widetilde{\rho}(0)$ and the preceding ones from the following ones. Hence the ordering of the crossings in the path plan is the order in which they occur along ρ, which is also their ordering in the wire plan. □

Wire plans are easy to compute. One simply walks down the elastic chain, choosing one crossing each time the chain crosses over a seam. Equivalently, for each segment of the chain one can identify the crossings of seams that occur within that segment (being careful not to duplicate the crossings that occur at joints), and concatenate them to obtain a path plan for the chain. Algorithm R does essentially this when constructing corridors for traces.

The elastic chains crossing a straight cut

Now we examine the more interesting aspects of elastic chains, namely, how they interact with cuts. We begin with a corollary of Lemma 7c.2.

Lemma 7c.4. *No two wires in a design have elastic chains that cross over, and no elastic chain for a wire crosses over itself.*

Proof. Let v and ω be (not necessarily distinct) wires in a design Ω; let κ and ρ be the elastic chains for v and ω, respectively. If κ and ρ cross over, then they have liftings $\widetilde{\kappa}$ and $\widetilde{\rho}$ that cross over. Let $\widetilde{v} \in [\widetilde{\kappa}]_P$ and $\widetilde{\omega} \in [\widetilde{\rho}]_P$ lift v and ω. Because $\widetilde{v}$ and $\widetilde{\omega}$ cohere, the endpoints of $\widetilde{\omega}$ lie on the same side of $\widetilde{v}$ (Lemma 4c.5). Hence by Lemma 7c.2, $\widetilde{\kappa}$ and $\widetilde{\rho}$ cannot cross over. □

The elastic-chain equivalent supports computation of link plans for all cuts that are not subpaths of the elastic chains. It can be made to deal with all cuts, but we will not need this extension. Let Φ be the **standard** elastic-chain equivalent of a design Ω, by which I mean the set of elastic chains for the wires in Ω, and let χ be a straight cut that is not a subpath of any elastic chain in Φ. First we need an ordering on crossings of χ by chains in Φ. Let (c, t) be a crossing of χ by $\rho \in \Phi$, and let (c', t') be a crossing of χ by $\rho' \in \Phi$. Let $\widetilde{\chi}$, $\widetilde{\rho}$, and $\widetilde{\rho}'$ lift χ, ρ, and ρ' to reflect the crossings (a, t) and (c', t'). Also let $\widetilde{\omega}$ be the unique lift of ω in $[\widetilde{\rho}]_P$. We say that (c, t) **precedes** (c', t') if and only if $\widetilde{\omega}$ separates $\widetilde{\chi}(0)$ from the endpoints of $\widetilde{\rho}'$. The **cut plan** of χ in Φ is the sequence of triples (ρ, t, c) denoting crossings of χ by chains in Φ, ordered by precedence. (The following proposition shows that the number of such crossings is finite.)

Proposition 7c.5. *Let Φ be the standard elastic-chain equivalent of a design Ω, and let χ be a straight cut that is not a subpath of any elastic chain in Φ. The cut plan of χ in Φ is akin to the path plans of χ in Ω.*

Proof. Let $\tilde{\chi}$ be any lift of χ, and let ρ be the rubber band of a typical wire $\omega \in \Omega$. Fix a particular path plan for χ in Ω.

The result hinges on a one-to-one correspondence between the crossings by ω in the path plan and the crossings of χ by ρ. Let (a,t) be a crossing of χ by ω corresponding to the lifting $\tilde{\omega}$ which separates the terminals of $\tilde{\chi}$. Let $\tilde{\rho} \in [\tilde{\omega}]_P$ lift ρ. Because the endpoints of $\tilde{\chi}$ lie on opposite sides of $\tilde{\rho}$, there is a crossing (c,r) of $\tilde{\chi}$ by $\tilde{\rho}$. Moreover, this crossing is unique. Suppose there were another, say (c',r'). Then $\tilde{\chi}_{c:c'}$ and $\tilde{\rho}_{r:r'}$ are both elastic, by Lemma 3d.2, and hence are identical (Lemma 3d.7). Assume without loss of generality that $[c,c']$ is a component of $\tilde{\chi}^{-1}(Im\,\tilde{\rho})$. Because ρ does not have χ as a subpath, either c or c' lies in $(0,1)$. Hence $\tilde{\rho}$ turns at a point in the middle of $\tilde{\chi}$, which is impossible, as the vertices of $\tilde{\rho}$ lie on $Bd\,M$. Thus $\tilde{\chi}$ and $\tilde{\rho}$ cross only at (c,r). Furthermore, the crossing (a,t) of χ by ω is akin to the crossing (c,r) of χ by ρ.

Now we show that every crossing (c,r) of χ by ρ can be obtained in this way. Given $\chi(c) = \rho(r)$, let $\tilde{\rho}$ be a lift of ρ satisfying $\tilde{\chi}(c) = \tilde{\rho}(r)$, and let $\tilde{\omega}$ be the corresponding lift of ω. I claim that $\tilde{\omega}$ separates the endpoints of $\tilde{\chi}$, and hence gives rise to a crossing of χ by ω in the path plan. For if not, then $\tilde{\rho}$, in intersecting $\tilde{\chi}$, would have to turn at a point in the middle of $\tilde{\chi}$; Lemma 7c.2 forbids it to cross over.

We have displayed a bijective correspondence between the crossing sequence of χ in Φ and the path plan of χ in Ω such that orresponding crossings are akin. It remains to show that corresponding crossings are in the same order in both sequences. This part should be clear, since the relation of precedence was expressly designed to make it work. □

Corollary 7c.6. *Let Φ be the any elastic-chain equivalent of a design Ω, and let χ be a straight cut that is not a subpath of any elastic chain in Φ. The cut plan of χ in Φ, after its trivial crossings are deleted, is akin to the link plans of χ in Ω.*

Proof. First take Φ to be the standard elastic-chain equivalent, and apply Proposition 7c.5. Link plans are obtained from path plans by deleting trivial crossings (Lemma 7b.3), and crossings that are akin are equally trivial or nontrivial. Now replace Φ by a different elastic-chain equivalent Ψ. What the cut plan of χ in Ψ becomes, when its trivial crossings are deleted, is something akin to the link plans of χ in an embedding Υ of Ω whose standard ECE is Ψ. Apply Lemmas 7b.2 and 7b.5 to get the result. □

Corollary 7c.6 gives us a handle on the content of the straight cut χ. From the cut plan of χ in Φ, we first delete the trivial crossings. We then replace each elastic chain in our plan by the wire that gave rise to it. This process must yield the content of χ. (When two crossing sequences are akin, each chain in one sequence has the same terminals as the corresponding chain in the other sequence, and the terminals of the elastic chain identify the wire it came from.) From the content we can derive the flows across the cut and many of its half-cuts, by Lemma 7b.4.

Determining precedence

Having computed elastic chains for all the wires in a design, one can incorporate them into a data structure like the rubber-band equivalent. To be useful, such a structure must help one compute which crossings of a cut by elastic chains are trivial, and it must also specify the precedence relation among the crossings. The problem of determining which crossings are trivial disappears in the sketch model, as I show in Section 8C, so I will not discuss it here. (See Chapter 10 for further discussion of this issue.) But the problem of computing the precedence relation appears in both models, and the same solution applies.

When two elastic chains cross a cut at the same point, we determine the precedence between the crossings by looking at where the two chains diverge. Suppose (c, s) and (c, t) are crossings of a straight cut χ by elastic chains σ and τ, respectively. (Possibly $\sigma = \tau$.) If these crossings are nontrivial, then σ crosses over χ at s, and similarly τ crosses over χ at t. Assume without loss of generality that both σ and τ cross over χ from left to right, looking from $\chi(0)$ toward $\chi(1)$. Now choose x and y as small as possible so that the subpaths $\sigma_{x:s}$ and $\tau_{y:t}$ coincide, segment for segment. (They may be parameterized differently.) I claim that (c, s) precedes (c, r) if and only if one of the following is true:

(1) $x > 0$ and $\sigma_{0:x}$ contacts τ from the right, or

(2) $y > 0$ and $\tau_{0:y}$ contacts σ from the left.

One of the two cases must apply. Let $\tilde{\chi}$, $\tilde{\sigma}$, and $\tilde{\tau}$ be liftings that reflect the crossings (c, s) and (c, t). The truth of the claim follows from Lemma 7c.3. Because σ and τ do not cross over, neither do $\tilde{\sigma}$ and $\tilde{\tau}$, and the side of $\tilde{\sigma}$ that contains the terminals of $\tilde{\tau}$ is the side from which $\tilde{\tau}$ contacts $\tilde{\sigma}$, or vice versa.

This characterization of precedence is depends on the cut χ only for an initial orientation. Consequently, the precedence relation can be represented in terms of orderings on the segments of elastic chains. It suffices to give a total ordering to the elastic chain segments that overlap. As in Section 1B, one can construct these orderings by adding one elastic chain at a time. When adding an elastic chain, one uses conditions (1) and (2) above to determine which previous chains lie immediately to its left and right. One can then insert the segments of the new chain between them. Precedence is transitive, so the orderings resulting from this process do indeed determine precedence among crossings.

Conformal embeddings

We turn now to a second structure for computing the plans of wires and the contents of cuts. Unlike the elastic-chain equivalent, it can only compute plans of wires with respect to a specific, built-in pattern. The seams of the pattern need not be straight, however. Given a design Ω and a pattern Γ, one computes an embedding Υ of Ω that conforms with Γ by successively finding and removing collapsible subpaths. Section 9B discusses this procedure in some depth; Proposition 7b.7 tells us that it works.

Once the "conformal" design Υ is at hand, one can read off the link plans of the wires in Υ and the cuts in Γ. Suppose the wire $v \in \Upsilon$ is link-homotopic to the wire $\omega \in \Omega$. The full plan of v in Γ is also its unique link plan in Γ, and hence (by Lemma 7b.2) that full plan is akin to every link plan for ω in Γ. Similarly, if χ is a cut of the design Ω, the content of χ in Ω can be recovered from the full plan of χ in Υ simply by replacing each wire v in this sequence by whichever wire $\omega \in \Omega$ is link-homotopic to v. This fact follows from Lemma 7b.5.

What is remarkable, however, is how the contents of cuts not in Γ can be computed using Γ and the conformal design Υ. Given a cut that conforms with Γ, one can find a link-homotopic link that conforms with Υ as well, and thereby compute the content of the original cut.

Proposition 7c.7. *Let the design Υ conform with the pattern Γ of nontrivial cuts, and let α be a link that conforms with Γ. Some link $\beta \in [\alpha]_L$ having the same roots in Γ as α conforms with both Γ and Υ.*

Proof. If α conforms with Υ, we are done. Otherwise we modify α until it conforms with Υ, maintaining its other properties as well. Suppose therefore that α does not conform with a wire $v \in \Upsilon$. First make α stable with respect to v: wherever α touches v without crossing over, or runs along v for some distance, displace it slightly. Because Γ is stable with respect to Υ, one can do so without introducing or removing any crossings of cuts in Γ by α, and without moving any of these crossings to an endpoint of α. In particular, the roots of α and the seam list of α in Γ remain unchanged, and α remains free in Γ.

If α still fails to conform with v, we modify α to obtain a better link β. By Lemma 7b.6, there is a deviation $\alpha_{s:t}$ across some subpath $v_{a:b}$. Recall the definition of deviation: either $\alpha_{s:t} \simeq_P v_{a:b}$, or else $\alpha_{t:s} \star v_{a:b}$ is a trivial link. In either case the link β defined by $\beta_{s:t} = v_{a:b}$ and $\beta(x) = \alpha(x)$ elsewhere is link-homotopic to α.

Now we show that β shares the important properties of α. First of all, β makes no more crossings with cuts in Γ than α does. If it did, then by rerouting $v_{a:b}$ to $\alpha_{s:t}$, one could eliminate some crossings of v in Γ. This cannot happen since v conforms with Γ. We conclude that β conforms with Γ just as α does. As a consequence, the link β is free in Γ because it conforms with Γ. Finally, β has the same roots as α. This could only fail if $\alpha(t)$ and $v(b)$ lay in different borders, so we can assume that the link $\alpha_{t:s} \star v_{a:b}$ is trivial. Let $\tilde{v}$ and $\tilde{\alpha}$ be liftings of v and α that reflect the crossing (a, s); they share a fringe F containing $\widetilde{\alpha}(t)$ and $\widetilde{v}(b)$. We prove that no Γ-lifting intersects F between those two points, whence their projections, the endpoints $\alpha(t)$ and $\beta(t)$ respectively, lie in the same border of the pattern Γ. Let (c, x) be the crossing of $\widetilde{v}$ by $\widetilde{\alpha}$ in which c is closest to b, so that the image of $\widetilde{v}_{c:b} \star \widetilde{\alpha}_{x:t}$ is a web of one thread. The points of F between $\widetilde{\alpha}(t)$ and $\widetilde{v}(b)$ are the only boundary points inside this web. Hence any Γ-lifting beginning in this portion of F would have to leave the web, since the cuts in Γ are nontrivial. Hence it would intersect either $\widetilde{\alpha}$ or $\widetilde{v}$, giving rise to an unnecessary crossing, and showing that either α or v does not conform with Γ.

Having produced the improved version β of α, we make it stable with respect to v and repeat the cycle. At each pass the number of crossings of wires in Υ by β decreases, so ultimately β is the desired link. □

We can restate Proposition 7c.7 in a weaker but more elegant form. Starting with any link χ, apply Lemma 4b.5 to find a link $\alpha \in [\chi]_L$ that conforms with the pattern Γ. Then Proposition 7c.7 produces a link $\beta \in [\chi]_L$ that conforms with both Γ and Υ.

Corollary 7c.8. *If a design Υ conforms with a pattern Γ of nontrivial cuts, then every link-homotopy class contains a cut that conforms with both Γ and Υ.* □

The real application of Proposition 7c.7, however, comes in Section 9B via the following result. It gives us a class of links in which to search for the link β guaranteed by Proposition 7c.7, and shows how the other links in this class can be rejected.

Corollary 7c.9. *Let the design Υ conform with the pattern Γ, and let α be a link that conforms with Γ. The content of α in Γ is the shortest wire list in Υ of a link β that is free in Γ and whose roots and seam list in Γ are those of α.*

Proof. Because α conforms with Γ, it is free in Γ. Hence Lemma 7a.9 applies to α and any link β satisfying the hypothesis; it shows them to be link-homotopic, whence by Lemma 7b.2 they have the same content. The content of a link β in Υ is always a subsequence of the wire list of β, and the two are equal precisely when β conforms with Υ.

Therefore it suffices to show that some link β that satisfies the hypotheses also conforms with Γ. Proposition 7c.7 gives us a link $\beta \in [\alpha]_L$ which conforms with both Γ and Υ (and hence is free in Γ); also β and α have the same roots in Γ. Finally, α and β both conform with Γ, and hence their seam lists in Γ are the sequences of cuts in their respective link plans. These sequences are equal, by Lemma 7b.2, because $\alpha \simeq_L \beta$. □

7D. The Geometry of Ideal Wires

To obtain a useful, constructive definition of ideal embeddings, we look more carefully at their geometry. So far our best characterization of an ideal wire is as a projection of a minimum-length path that avoids its forbidden zones. This definition can probably be converted into a polynomial-time algorithm. For the sake of efficiency and simplicity, however, we must avoid dealing directly with zones in a blanket. Because ideal wires are taut, their shapes are powerfully constrained. We can use these constraints to build up ideal wires from simpler pieces, as in Algorithm R.

In the remainder of this chapter I show how to derive the ideal embedding of a wire from a simple geometric specification: a set of corridors called a *maze*. This section abstracts the important properties of ideal wires, and defines the maze through which an ideal wire is routed. Section 7E then shows how to reconstruct the ideal wire from its maze. The topological inputs to this process are the path plans of the ideal wire in certain patterns and link plans for the cuts in those patterns. These sequences, along with the endpoints of the ideal wire, are assumed to be known. Due to technical difficulties, the construction does not apply to all ideal wires, but it applies to all the wires we derive from sketches in Chapter 8. Consequently, the routing method carries over to the sketch model, where the plans can be efficiently supplied by the rubber-band equivalent and trace endpoints are fixed. In that model it reduces to Algorithm R.

The segments and struts of an ideal wire

One thing we know about ideal wires is that they are taut and therefore tangent to their barriers. A few definitions help to clarify this relationship. Recall that a **segment** of a piecewise linear path α is a maximal linear subpath $\alpha_{s:t}$ of α with $s < t$. Let ω be a piecewise linear path, and let σ be a straight path ending at $\alpha(t)$. If ω turns toward $\sigma(0)$ at t, then σ **supports** ω at t. If $\omega_{s:t}$ or $\omega_{t:s}$ is a segment of ω, then we also say that σ supports this segment. For example, if σ is a strut for an ideal wire ω, and $\sigma(1)$ is not an endpoint of ω, then σ supports two segments of ω. A straight path α in R^2 is **tangent** to a straight path σ if the line containing α intersects the polygon

$$P(\sigma) = \{\, x : \|x - \sigma(0)\| = \|\sigma\| \,\}$$

at $\sigma(1)$, but does not intersect its inside. A straight path σ in a sheet S is **diagonal** if $\sigma(1)$ is a vertex of $P(\sigma)$ (in the terminology of Section 1D, the slope of σ is diagonal), and $\sigma(0)$ is a vertex of a fringe of S. These definitions are contrived for the purpose of stating the following important lemma.

Lemma 7d.1. *Every strut for an ideal wire is diagonal, and each segment of an ideal wire is tangent to the struts that support it.*

Proof. Let ω be an ideal wire in a sheet S; let M be the blanket of S, and $p\colon M \to S$ the covering map. Suppose τ be a strut for an ideal wire ω at t. Lift τ and ω to $\tilde{\tau}$ and $\tilde{\omega}$ so that $\tilde{\tau}(1) = \tilde{\omega}(t)$. Because τ is a strut, $\tilde{\omega}$ turns toward $\tilde{\tau}(0)$ at t. And because struts are marginal and nondegenerate, any straight half-link κ with $\kappa(0) = \tilde{\tau}(0)$ and $\|\kappa\| < \|\tau\|$ is forbidden to $\tilde{\omega}$. All points sufficiently close to $\tilde{\omega}(t)$ are visible from $\tilde{\tau}(0)$. Thus in the neighborhood of $\tilde{\omega}(t)$, all points within $\|\tau\|$ units of $\tilde{\omega}(t)$ are forbidden. Since $\tilde{\omega}$ is evasive, it avoids these points, and hence the segments of ω supported by τ stay at least $\|\tau\|$ units from $\tau(0)$. But they do intersect $\tau(1)$, so they are tangent to τ.

An extension of this argument shows that τ is diagonal. Let Q be a barrier for $\tilde{\omega}$ containing the forbidden half-links $\tilde{\tau}_{0:x}$ for $x \in (0,1)$. Because $\tilde{\omega}$ turns at t, the barrier Q has a convex corner at t. In a neighborhood of $\tilde{\omega}(t)$, the set Q contains all points closer to $\tilde{\tau}(0)$ than $\tilde{\tau}(1)$. Hence $\tau(1)$ lies on a vertex, not an edge, of the polygon $P(\tau)$. Again, all half-links obtained by translating $\tilde{\tau}_{0:x}$ a sufficiently small distance along the base of Q contribute to Q. If $\tau(0)$ lay on a fringe edge, then Q would not have a convex corner at $\tilde{\tau}(1)$. So $\tau(0)$ is a fringe vertex, and therefore τ is diagonal. □

A pleasant consequence of Lemma 7d.1 is that an ideal wire has exactly one strut at each joint. Given that the strut is diagonal, its slope is determined by the segments that it supports. This fact will be clarified shortly.

Representation of angles

Lemma 7d.1 shows that the shape of an ideal wire is heavily influenced by the wiring norm. It is therefore convenient to relate angles and vectors in an ideal design to a standard geometric representation of the wiring norm: its **unit polygon** $C = \{\, x : \|x\| = 1 \,\}$, the set of vectors of norm 1. We represent angles, or directions, by points of C. If δ and θ are points of C, then an interval such as (δ, θ) denotes the points of C lying between δ and θ in clockwise order. The **angle** at which a path σ travels is denoted $\dot{\sigma}$, and defined as

$$\dot{\sigma} = \frac{\sigma(1) - \sigma(0)}{\|\sigma(1) - \sigma(0)\|},$$

provided that σ is not a loop. Normally σ will be straight. The vertices Δ of C are called **diagonal angles**. A straight path σ in a sheet is diagonal if and only if $\dot{\sigma} \in \Delta$ and $\sigma(0)$ is a fringe vertex.

Whether one path is tangent to another depends only on their angles and where they intersect. For each diagonal angle $\delta \in C$, let $\delta^{\perp}$ and $\delta^{\top}$ denote the angles

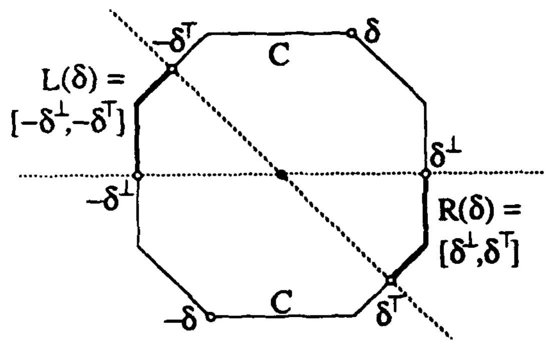


Figure 7d-1. *Angles as points on the unit polygon.* Being a vertex of the unit polygon C, the point δ is a diagonal angle. Any straight path tangent to a path σ of angle δ has angle in one of the closed intervals $R(\delta)$ and $L(\delta)$ (dark borders), depending on whether it leaves σ to the right or the left.

of the segments of C preceding and following δ, oriented clockwise. Let α and σ be straight paths, and suppose $\alpha(t) = \sigma(1)$. The path α is tangent to σ leaving $P(\sigma)$ to its right if and only if $\dot\alpha$ lies in the interval $R(\dot\sigma) = [\dot\sigma^{\perp}, \dot\sigma^{\mathsf T}]$. Similarly, α is tangent to σ leaving $P(\sigma)$ to its left if and only if $\dot\alpha \in L(\dot\sigma) = [-\dot\sigma^{\perp}, -\dot\sigma^{\mathsf T}]$. The polygon C has inversion symmetry, and so $(-\delta)^{\perp} = -(\delta^{\perp})$ and $(-\delta)^{\mathsf T} = -(\delta^{\mathsf T})$. In other words, the operator $(-)$ commutes with $(^{\perp})$ and $(^{\mathsf T})$. One corollary is that $L(\dot\sigma) = R(-\dot\sigma)$.

Tracks and ties

With a view toward the sketch model, we now abstract away from struts and ideal wires. Recall that a **joint** of a piecewise linear path ω is a point $s \in (0,1)$ at which ω is not linear. A piecewise straight path ω is a **track** if for every joint s of ω, there is a straight path σ supporting ω at s whose angle is diagonal, and the two segments of ω supported by σ are tangent to σ. We call σ a **tie** for ω at s. Lemma 7d.1 implies that an ideal wire is a track, and its struts are ties.

The angles of the ties of a track ω are determined by ω. Let σ support a track ω at t; by symmetry we may assume that ω leaves σ to the right. Then σ supports two segments of ω. Because those two segments are tangent to σ, their angles lie in $R(\dot\sigma)$. If δ and θ are distinct diagonal angles, then the intervals $R(\delta)$ and $R(\theta)$ intersect in at most one point. The two segments of ω supported by σ have different angles, and so these angles determine a unique interval $R(\dot\sigma)$. If ω is an ideal wire, then the angle $\dot\sigma$ determines a unique strut σ for ω at t.

Angles of consecutive ties

Next we consider how the angles of ties vary as one moves along a track. Let Δ be the set of diagonal angles, the vertices of the unit polygon C. If δ is an angle in Δ, let $cw(\delta)$ denote the next angle clockwise in Δ, and let $ccw(\delta)$ denote the next angle counterclockwise in Δ.

Lemma 7d.2. *Let σ and τ support a track ω at s and t, respectively, and suppose $\omega_{s:t}$ is a segment of ω.*

(1) *If σ and τ lie on opposite sides of ω, then $\dot\tau = -\dot\sigma$.*
(2) *If σ and τ lie to the right of ω, then $\dot\tau \in \{\dot\sigma, cw(\dot\sigma)\}$.*
(3) *If σ and τ lie to the left of ω, then $\dot\tau \in \{\dot\sigma, ccw(\dot\sigma)\}$.*

Proof. By symmetry we may assume that ω leaves σ to the right. Let α denote the segment $\omega_{s:t}$. Lemma 7d.1 says that α is tangent to σ, which implies $\dot\alpha \in [\dot\sigma^{\perp}, \dot\sigma^{\top}]$. Let κ be the segment of ω preceding α; it exists because $s > 0$. Then $\dot\kappa \in [\dot\sigma^{\perp}, \dot\sigma^{\top}]$ also, by Lemma 7d.1, and $\dot\alpha$ lies clockwise of $\dot\kappa$. Therefore $\dot\alpha \in (\dot\sigma^{\perp}, \dot\sigma^{\top}]$. If ω leaves τ to the right, then we also have $\dot\alpha \in [\dot\tau^{\perp}, \dot\tau^{\top}]$. Now let κ be the segment of ω following α; it exists because $t < 1$. Then $\dot\kappa \in [\dot\tau^{\perp}, \dot\tau^{\top}]$ also, and $\dot\alpha$ lies counterclockwise of $\dot\kappa$. Thus $\dot\alpha \in [\dot\tau^{\perp}, \dot\tau^{\top})$. The only way the intervals $(\dot\sigma^{\perp}, \dot\sigma^{\top}]$ and $[\dot\tau^{\perp}, \dot\tau^{\top})$ can intersect is if $\dot\tau$ is either $\dot\sigma$ or $cw(\dot\sigma)$. This establishes conclusion (2). Now suppose that ω leaves τ to the left. Then by the same kind of reasoning, $\dot\alpha$ lies in $(-\dot\tau^{\perp}, -\dot\tau^{\top}]$. The only way this interval can intersect $(\dot\sigma^{\perp}, \dot\sigma^{\top}]$ is if $\dot\tau = -\dot\sigma$. This proves conclusion (1). Part (3), and the case of part (1) in which ω leaves σ to the left, follow by symmetry. □

Lemma 7d.2 is the key to the behavior of tracks. It implies, among other things, that the angles of the a track's ties change incrementally as one moves along the track. If σ supports a track ω, we write $\ddot\sigma$ for $\dot\sigma$ or $-\dot\sigma$ according to whether ω leaves σ to the right or the left. Lemma 7d.2 implies that if σ and τ are consecutive ties of ω, then $\ddot\tau \in \{\ddot\sigma, cw(\ddot\sigma), ccw(\ddot\sigma)\}$.

Subpaths of a track

The prime consequence of Lemma 7d.2 concerns the angles at which subpaths of a track travel. If $\delta \in \Delta$, the ties of angle $\pm\delta$ for a track ω divide ω into subpaths that I call **δ-subpaths**. A path $\omega_{s:t}$ with $s < t$ is a δ-subpath of ω if

- either $s = 0$ or ω has a tie of angle $\pm\delta$ at s, and
- either $t = 1$ or ω has a tie of angle $\pm\delta$ at t.

Lemma 7d.3. *Let $\omega_{s:t}$ be a δ-subpath of a track ω, and suppose $\omega_{s:t}$ is not straight. For some $\theta = \pm\delta$, every path $\alpha = \omega_{x:y}$ with $s \le x < y \le t$ satisfies $\dot\alpha \in [\theta^{\top}, -\theta^{\perp}]$, and $\dot\alpha \in (\theta^{\top}, -\theta^{\perp})$ if $[x,y] = [s,t]$. If σ supports ω at s, then $\theta = \dot\sigma$, and if τ supports ω at t, then $\theta = -\dot\tau$.*

Proof. Let κ support ω at a point $x \in (s,t)$. By Lemma 7d.2, $\ddot\kappa$ changes by at most one step clockwise or counterclockwise as one moves along $\omega_{s:t}$. Since $\ddot\kappa$ is never $\pm\delta$, it is trapped in some interval $(\theta, -\theta)$ where $\theta = \pm\delta$. Put $N = [\theta^{\top}, -\theta^{\perp}]$. We show that every segment α of $\omega_{s:t}$ satisfies $\dot\alpha \in N$. Each such segment is supported by a tie κ with $\ddot\kappa \in (\theta, -\theta)$. Lemma 7d.1 says that α is tangent to κ, which means $\dot\alpha \in R(\ddot\kappa)$

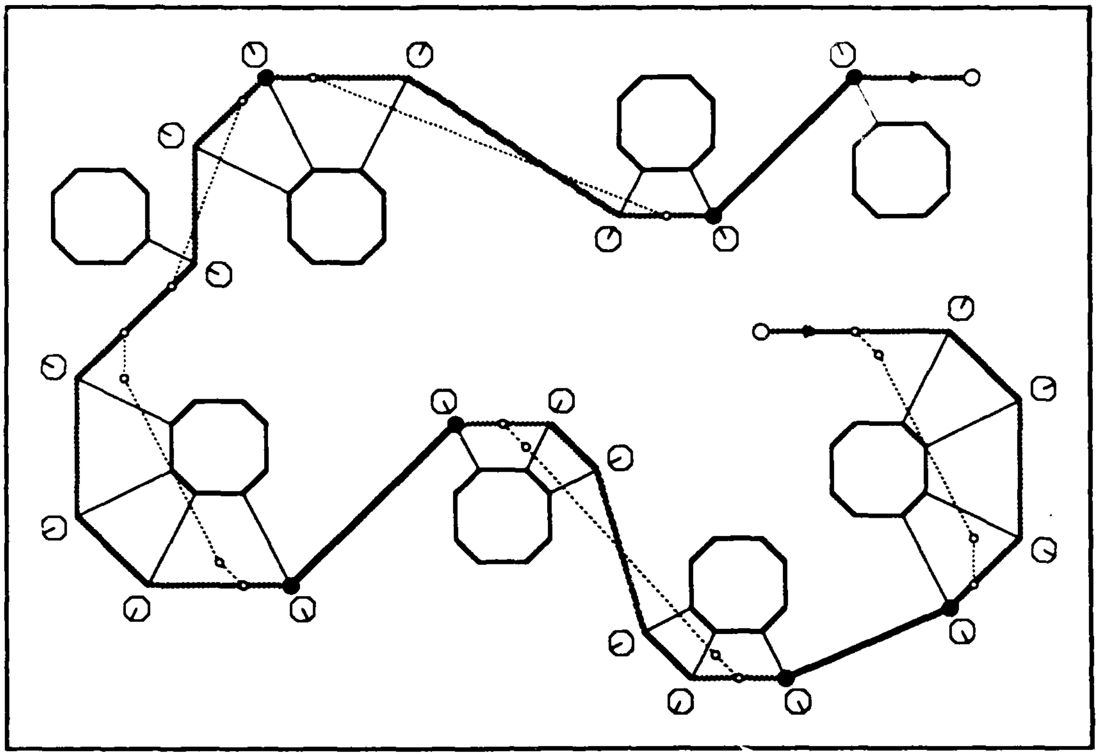


Figure 7d-2. *A track and its ties.* The grey and black segments compose a track under the octagonal wiring norm of Figure 7d-1. Ties for this track are shown as thin lines. Near each tie σ is a small picture of the unit polygon with a tick mark to indicate the diagonal angle $\ddot{\sigma}$. This angle moves by at most one vertex from one tie to the next. One diagonal angle δ (roughly, north-northwest) is singled out. Joints corresponding to ties of this angle are shown as black dots; they are the endpoints of the track's "δ-subpaths". Black segments are its straight δ-subpaths. The dotted path, whose joints are shown as small circles, is a "δ-route" for the track, as discussed in Section 7E. It is shown only where it does not overlap with the track.

if ω leaves κ to the right, and $\dot{\alpha} \in L(-\ddot{\kappa})$ if ω leaves κ to the left. In either case $\dot{\alpha}$ lies in the interval $[\ddot{\kappa}^{\top}, -\ddot{\kappa}^{\perp}]$, which is contained in N when $\ddot{\kappa} \in (\theta, -\theta)$. Therefore $\dot{\alpha} \in N$ as desired. Another elementary fact is that the sum of two vectors whose angles lie in N has angle in N. Hence if $s \leq x < y \leq t$, then the path $\alpha = \omega_{x:y}$ satisfies $\dot{\alpha} \in N$ by induction on the number of segments of α. Moreover, because $\omega_{s:t}$ turns, it contains segments of angle strictly between $\theta^{\top}$ and $\theta^{\perp}$. Hence if $s = x$ and $y = t$, then $\dot{\alpha}$ lies in the interior of N.

It remains to consider the relationship between θ and the ties, if any, that support ω at s and t. Let σ be a tie of angle $\pm\delta$ supporting ω at s, and suppose first that ω

leaves σ to the right. Then Lemma 7d.2 implies that the next tie κ after σ satisfies $\dot{\kappa} = cw(\dot{\sigma})$ and lies right of ω, since $\dot{\kappa} \neq \pm\delta = \pm\dot{\sigma}$. Therefore $\ddot{\kappa} = \dot{\kappa} \in (\theta, -\theta)$ only if $\theta = \dot{\sigma}$ as claimed. Now suppose ω leaves σ to the left. Then by the same reasoning, $\dot{\kappa} = ccw(\dot{\sigma})$ and ω leaves κ to its left. Hence $\ddot{\kappa} = -\dot{\kappa} = ccw(-\dot{\sigma})$, and so $\ddot{\kappa} \in (\theta, -\theta)$ only if $-\theta = -\dot{\sigma}$ as claimed. Now let τ be a tie of angle $\pm\delta$ supporting ω at t. Symmetrical reasoning shows that $\theta = -\dot{\tau}$. □

Cuts that contain struts

Now that we understand some intrinsic properties of tracks, we return to ideal wires. Lemma 7d.2 leads us to an important fact about the cuts that contain struts.

Lemma 7d.4. *Let σ be a strut for an ideal wire ω at s, and let χ be the straight link of which σ is a subpath; say $\sigma = \chi_{0:a}$. Then the crossing (a, s) is similar to no other crossing of χ by ω.*

Proof. Let $\widetilde{\omega}$ and $\widetilde{\chi}$ be lifts of ω and χ satisfying $\widetilde{\chi}(a) = \widetilde{\omega}(s)$. Let $\widetilde{\sigma}$ be the lift $\widetilde{\chi}_{0:a}$ of σ. Because σ is a strut, it is nondegenerate, and hence $\widetilde{\chi}(0)$ does not lie on a terminal of $\widetilde{\omega}$. Furthermore, $\widetilde{\omega}$ crosses over $\widetilde{\chi}$ at s. If another crossing of χ by ω is similar to (a, s), then $\widetilde{\omega}$ crosses back at some point at some point $\widetilde{\omega}(t) = \widetilde{\chi}(b)$, where $b > a$. We derive a contradiction from this assumption. We may assume that ω leaves σ to its right.

If t is chosen as close to s as possible, then the loop $\lambda = \widetilde{\omega}_{s:t} \star \widetilde{\chi}_{b:a}$ is simple. Let N be the inside of this loop. By Lemma 3c.6, the internal angles of λ sum to $(n-2)\pi$, where n is the number of vertices of λ. Hence as one travels around λ, the angles of the segments of λ must rotate through a total of 2π in the direction of N (in this case, counterclockwise). Not all of this rotation can occur at the points $\widetilde{\chi}(a)$ and $\widetilde{\chi}(b)$. Equivalently, let us associate with each vertex $\widetilde{\omega}(x)$ of λ for $s \leq x < t$ the angle $\ddot{\tau}$ derived from the strut τ for ω at $\omega(x)$. This angle begins at $\ddot{\sigma}$, and must rotate counterclockwise past $\ddot{\sigma}$. Furthermore, if $\ddot{\tau}$ is thought of as a vector based at $\widetilde{\omega}(x)$, it always points into $Cl\,N$.

Lemma 7d.2 implies that as one travels along $\widetilde{\omega}$, the angles $\ddot{\tau}$ change by at most one step clockwise or counterclockwise for each joint visited. Hence there must be a point x in (s, t) whose associated angle is $\ddot{\sigma}$, and such that the next associated angle is counterclockwise from this. Again by Lemma 7d.2, this can only happen if ω has a strut τ on its left at x. If $\widetilde{\tau}$ is the lift of τ with $\widetilde{\tau}(1) = \widetilde{\omega}(x)$, then $\widetilde{\tau}$ intersects N. And here we obtain our contradiction. For the lift $\widetilde{\tau}$ is parallel to $\widetilde{\sigma}$ and cannot leave $Cl\,N$ via the middle of $\widetilde{\chi}_{a:b}$. It follows that $\widetilde{\tau}$ must leave N at some point in $Im\,\widetilde{\omega}_{s:t}$. But because τ is a strut for ω, the points $\widetilde{\tau}(z)$ for $z < 1$ are forbidden to $\widetilde{\omega}$. Therefore $\widetilde{\omega}$ is not evasive, a contradiction. □

Tunnels and mazes

The final result of this section relates an ideal wire to a set of corridors. Before explaining this relation, however, we replace the concept of corridor with a more technically convenient one. The new concept is more restrictive, and it uses straight paths (*gates*) in place of the line segments (*doorways*) that form a corridor.

Definition 7d.5. A **tunnel** is a finite sequence $\langle \lambda_0, \lambda_1, \ldots, \lambda_m, \lambda_{m+1} \rangle$ of linear paths in the plane, of which the first and last are constant, together with a sequence $0 = t_0 < t_1 < \cdots < t_m < t_{m+1} = 1$ such that

(1) no two consecutive paths λ_i and λ_{i+1} intersect, and

(2) for $1 \leq i \leq m$, there is a line L of slope $\pm\delta$ containing $Im\ \lambda_i$ such that neither open half-plane of L contains both $Im\ \lambda_{i-1}$ and $Im\ \lambda_{i+1}$.

For $1 \leq i \leq m+1$, the path γ_i is called a **gate at parameter** t_i. If all the lines L in (2) can be chosen to have angle $\pm\delta$, then the tunnel is a **δ-tunnel**.

Recall that Algorithm R builds several corridors for each wire, one per diagonal slope. (A diagonal slope corresponds to a pair $\pm\delta$ of diagonal angles.) In the new terminology, these corridors become the tunnels in a **maze**. A maze is a set of tunnels which have the same initial and final gates: one δ-tunnel for each diagonal slope δ.

What tunnels and mazes do is specify a set of paths. A path **through** the tunnel in Definition 7d.5 is any path $\alpha: I \rightarrow R^2$ such that $\alpha(t_i) \in Im\ \lambda_i$ for $0 \leq i \leq m+1$. A path through a maze is any path through all the tunnels in the maze. Minimum-length paths through tunnels and mazes are of particular interest to us, and can be characterized by a simple geometric condition called *tightness*. A gate λ **restrains** a PL path α at a point $t \in (0,1)$ if $\alpha(t)$ is an endpoint of λ and either λ is constant or α turns away from the other endpoint of λ at t. A PL path α through a tunnel (or maze) is **tight** in that tunnel (or maze) if for every joint t of α, the tunnel (or maze) contains a gate with parameter t that restrains α at t. Proposition 7e.1 shows that the minimum-length path through a tunnel is, up to parameterization, the unique tight path through the tunnel.

A maze for an ideal wire

I now offer a construction which, for most ideal wires, leads to a maze in which that wire is tight. As we show in Section 8C, it applies to all ideal embeddings of designs derived from sketches. The maze for such a wire can be easily computed from the elastic-chain equivalent of the design, and as we show in the Section 7E, the ideal wire can be reconstructed from the maze. The relation between the ideal wires and their mazes will be used in Section 8C to explain Algorithm R.

To define the maze for an ideal wire ω, we first need a special collection of patterns. For pair $\pm\delta$ of diagonal angles, we assume we are given a pattern Γ with three properties.

(1) Every cuts in Γ has angle $\pm\delta$.

(2) Every strut for ω is a subpath of some cut in Γ.

(3) For every seam $\gamma \in \Gamma$, the line containing γ separates the interiors of the pieces of Γ that include $Im\,\gamma$.

The δ-tunnel in the maze for ω is defined as follows. Its first gate is $\omega_{0:0}$, its last gate is $\omega_{1:1}$, and it has one intermediate gate for each element of a path plan for ω in Γ. Suppose the ith element of this plan is a crossing (c, t) of a cut $\gamma \in \Gamma$ by ω. We may assume that t increases with i. Let A and B denote the terminals of γ, and define a and b by the equations

$$\|\gamma_{0:a}\| = \mathit{flow}(\gamma_{0:c}, \Omega) + \mathit{width}(A)/2 + \mathit{width}(\omega)/2, \tag{7–6}$$

$$\|\gamma_{1:b}\| = \mathit{flow}(\gamma_{1:c}, \Omega) + \mathit{width}(B)/2 + \mathit{width}(\omega)/2. \tag{7–7}$$

If the half-cut $\chi_{0:c}$ for ω at r is trivial, however, then put $a = 0$, and put $b = 1$ if $\chi_{1:c}$ is trivial. The ith gate is the path $\gamma_{a:b}$ at parameter t. By Lemma 7b.2, all path plans for ω in Γ are akin and include the same sequence of cuts, the path plan we choose does not affect the gates in the tunnel, but only affects the parameters t.

One can easily check that these gates form a δ-tunnel, using the fact that the sequence of cuts in the path plan of ω is also the path code of ω. Condition (1) in Definition 7d.5, namely that consecutive gates must be disjoint, holds because the seams in Γ are disjoint and no seam of Γ appears twice consecutively in the path plan. Condition (2) in Definition 7d.5 follows from property (3) of Γ, since consecutive seams in the path plan belong to the same piece of Γ.

Building the maze

Excepting the parameterization information, which is not important in the applications, and excepting the first and last gates, the δ-tunnel for the ideal wire ω can be constructed without knowing ω. All one needs is a way of computing the path plan of ω in the pattern Γ, or something akin to it, and a way of locating the crossings in this plan within the path plans of the cuts in Γ, in order to compute the flows across the half-cuts in equations (7–6) and (7–7). In what follows, we let Ω be the design and Γ the pattern for the δ-route of ω.

One way is to use an elastic-chain equivalent Φ of Ω that contains the chain ρ of ω. By Lemma 7b.2, every path plan of ρ in the pattern Γ is akin to the path plans of ω in Γ, and the sequence of cuts in the two plans is the same. Fix path plans for ρ and ω in Γ. Let the ith crossing of the path plan for ρ be the crossing (γ, d, r)

of $\gamma \in \Gamma$, and let the ith crossing in the path plan for ω be the crossing (γ, c, t). These two crossings are akin. Take the interesting case where the crossing (d, r) is nontrivial, and suppose that the triple (ρ, r, d) is the kth crossing out of n in the link plan of γ in Φ. We can apply Corollary 7c.6 to this link plan: Since ρ crosses over γ, the straight cut γ cannot be a subpath of any elastic chain in Φ, by Lemma 7c.4. By Corollary 7c.6, the link plan for γ in Φ is akin to every link plan for γ in the design Ω, and in particular to one that includes the crossing (ω, t, c) akin to (ρ, r, d). Hence from the link code of γ in Φ we can derive the link code $\omega_1, \ldots, \omega_n$ of γ in Ω, simply by replacing each elastic chain with the wire that gave rise to it. We also know that in some link plan for γ in Ω, the triple (ω, t, c) is the kth crossing. Hence by Lemma 7b.4 we have

$$\mathit{flow}(\gamma_{0:c}, \Omega) = \sum_{i=1}^{k-1} \mathit{width}(\omega_i) \qquad \text{and} \qquad \mathit{flow}(\gamma_{1:c}, \Omega) = \sum_{i=k+1}^{n} \mathit{width}(\omega_i).$$

Combining these equations with (7–6) and (7–7), we get

$$\|\gamma_{0:a}\| = \mathit{width}(A)/2 + \mathit{width}(\omega_k)/2 + \sum_{i=1}^{k-1} \mathit{width}(\omega_i); \tag{7–8}$$

$$\|\gamma_{1:b}\| = \mathit{width}(A)/2 + \mathit{width}(\omega_k)/2 + \sum_{i=k+1}^{n} \mathit{width}(\omega_i). \tag{7–9}$$

(Here A and B are the terminals of γ.) Equations (7–7) and (7–8) tell us that the ith gate in the δ-tunnel for ω is $\gamma_{a:b}$. Gates derived from trivial crossings can be handled in the same framework.

Another method, which only generates the gates corresponding to nontrivial crossings, starts from an embedding Υ of Ω that conforms with the pattern Γ. If $v \in \Upsilon$ is the wire that embeds ω, then the full plan of v in Γ is also its unique link plan, and hence is akin to the link plans for ω in Γ. As Lemma 7b.3 shows, every such link plan consists exactly of the nontrivial crossings in a path plan for ω in Γ. At the same time, those crossings (or rather, their reverses) are part of the full plans in Υ of the cuts in Γ, which also equal the link plans of those cuts. Hence Lemma 7b.4 applies as above to determine the gates in the δ-tunnel of ω, or at least those corresponding to nontrivial crossings. In the application of this method to the sketch model, the gates derived from trivial crossings can be ignored.

Tightness of the ideal wire

The following result is our geometric characterization of ideal wires. It is also the precondition for the reconstruction process of the next section.

Proposition 7d.6. *An ideal wire is a tight track through its maze.*

Proof. Lemma 7d.1 shows that the ideal wire ω is a track. To prove that ω passes through its maze, it suffices to show that ω passes through its δ-tunnel for arbitrary δ. We adopt the notation of equations (7–6) and (7–7), so the ith gate in this tunnel is $\gamma_{a:b}$ at parameter t, and $\omega(t) = \gamma(c)$. We need only prove $a \leq c \leq b$. If $\chi_{0:c}$ is a nontrivial half-cut, then its flow does not exceed its capacity because ω is evasive. Thus the right-hand side of equation (7–6) above is at most $\|\chi_{0:c}\|$, and $\|\chi_{0:a}\| \geq \|\chi_{0:c}\|$ implies $a \leq c$. Or if $\chi_{0:c}$ is trivial, then $c \geq a = 0$. Similarly, equation (7–7) implies $c \leq b$ if $\chi_{1:c}$ is nontrivial, and if $\chi_{1:c}$ is trivial, then $c \leq b = 1$.

It remains to show that ω is tight in its maze. Supposing that t is a joint of ω, we show that ω has a gate at t that restrains ω at t. Because ω is taut, it has a strut σ at t, and by Lemma 7d.1, the angle $\delta = \dot{\sigma}$ is diagonal and the point $\sigma(0)$ is a fringe vertex. Let Γ be the pattern for S from which the δ-tunnel for ω was derived. Then σ is a subpath of a unique diagonal cut $\gamma \in \Gamma$. Assume that $\sigma = \gamma_{0:c}$; the other case $\sigma = \gamma_{1:c}$ is symmetrical. Apply Lemma 7d.4 with γ in place of χ and (c, t) in place of (a, s). It says that if $\widetilde{\gamma}$ and $\widetilde{\omega}$ are lifts that reflect (c, t), then they make no other crossings. Because σ is a strut, $\widetilde{\omega}$ crosses over $\widetilde{\gamma}$ at t. Hence $\widetilde{\gamma}$ separates the endpoints of $\widetilde{\omega}$. We conclude that (c, t) appears in any path plan for ω in Γ. Consider the corresponding gate $\gamma_{a:b}$ of the δ-tunnel. Because σ is marginal, and hence $\sigma = \gamma_{0:a}$ in equation (7–6). Because σ is nontrivial, the crossing (c, t) is nontrivial. Therefore $c = a$, and $\gamma_{a:b}$ begins where σ *ends. Finally, since* σ supports ω at t, the gate $\gamma_{a:b}$ restrains ω at t. □

7E. Routing Through a Maze

Now we return to the abstract setting of tracks and ties in order to justify a general routing method. Given a maze in which a track is tight, we prove give a general procedure for reconstructing the track from the maze. The shortest path, in euclidean arc length, through a tunnel of a maze is called a **partial route** for the maze. No partial route by itself need by a path through the maze, but they can be combined into a tight track through the maze if any such track exists. This result implies that an ideal wire may by efficiently constructed if its endpoints are known. More importantly, it allows us to prove in Section 8C the correctness of Algorithm R.

The intuition behind our construction is the following. Starting from a track and a maze in which it is tight, we remove all the gates restraining the track except those of a particular diagonal slope $\pm\delta$. The track then relaxes to one of its partial routes. Compare Figure 7d-2. At a joint of the track supported by a tie of angle $\pm\delta$, the path does not move. The segments incident on that joint used to be tangent to the tie, and after relaxation they may make more acute angles with the tie than before, but they cannot make more obtuse angles than before. Similarly, the straight δ-subpaths of the track do not move. On the other hand, consider a new joint formed during relaxation. Such joints can only be formed within parts of the track that used to be nonstraight δ-subpaths. By Lemma 7d.3, all segments of those δ-subpaths have angles that make them ineligible for tangency with ties of angle $\pm\delta$. As the track relaxes to its partial route, those segments cannot become tangent to ties of angle $\pm\delta$; one of the two segments incident on a new joint always makes too obtuse an angle. (We prove this.) Hence we can distinguish the joints of the partial route that are *retained* in the track by the angles of the incoming segments. Similarly, we can determine which segments of the partial route are retained in the track by looking at their angles; they have to be tangent to ties of angle $\pm\delta$. Using some information from Lemma 7d.2 about how the track is allowed to turn, we can combine the retained joints and segments to build the track.

Partial routes

Our source of information on partial routes is the following crucial result. Two piecewise linear paths $\alpha, \beta: I \to R^2$ are **alike** if they turn at the same points, and whenever α and β turn at $t \in (0,1)$ we have $\alpha(t) = \beta(t)$.

Proposition 7e.1. *A path through a tunnel has minimum euclidean arc length if and only if it is tight. All tight paths through a tunnel are alike.* □

The proof of Proposition 7e.1 is not trivial. I omit it only because essentially the same arguments that led up to Corollary 3d.7 apply here.

Restricted routes

To relate the partial routes in a maze to a tight track through the maze, we introduce *restricted routes* that are more closely tied to the track. Later we prove the restricted and partial routes equal, thus obtaining more information about the latter. Let ω be a tight track through a maze, and let δ be a diagonal angle. The partial route through the δ-tunnel of this maze is called the **δ-route** of ω. The **restricted δ-route** of ω is the shortest path through a restricted version of the maze's δ-tunnel. The restricted tunnel also begins at $\omega(0)$ and ends at $\omega(1)$, and its ith gate is derived from the ith gate in the δ-tunnel, say γ at parameter t. If γ restrains the δ-route at the point t where it passes through γ, then the ith gate is the constant path at $\omega(t)$. Otherwise the ith gate is just γ.

Any path through the restricted δ-tunnel is also a path through the original δ-tunnel. Thus the restricted δ-route, call it ρ, is just like the δ-route of ω except that it has to pass through the endpoints of every tie for ω of angle $\pm\delta$. The path ω passes through its restricted δ-tunnel.

Lemma 7e.2 is the main result concerning restricted routes.

Lemma 7e.2. *Let ρ be the restricted δ-route of a track ω. For each δ-subpath $\omega_{s:t}$ of ω, either*

(1) *$\omega_{s:t}$ is straight and equals $\rho_{s:t}$, or*

(2) *each segment α of $\rho_{s:t}$ satisfies $\dot{\alpha} \in [\theta^{\top}, -\theta^{\perp}]$, where θ given by applying Lemma 7d.3 to $\omega_{s:t}$.*

Proof. The path ω has either endpoints or ties of angle $\pm\delta$ at s and t, and hence $\rho(s) = \omega(s)$ and $\rho(t) = \omega(t)$. If $\omega_{s:t}$ is straight, then it must equal $\rho_{s:t}$ since ω is a path though the tunnel that defines ρ and straight paths are always minimum-length. So we may assume $\omega_{s:t}$ is not straight and prove case (2). Naturally, our primary tool will be Lemma 7d.3, which defines supporting angles. That lemma gives us an angle $\theta \in \{\delta, -\delta\}$ such that every subpath $\alpha = \omega_{x:y}$ with $s \le x < y \le t$ satisfies $\dot{\alpha} \in [\theta^{\top}, -\theta^{\perp}]$.

A subgoal is to prove that the angles of the segments of $\rho_{s:t}$ never approach θ. First we show if one segment of ρ has angle in $(\delta, -\delta)$, then the next has angle in $[\delta, -\delta]$. Thinking of δ as "north", this claim says ρ cannot switch from pointing east to pointing west without pointing exactly north or south in between. The reason is that *ρ is a minimum-length path through a δ-tunnel*. By the definition of δ-tunnel (see Definition 7d.5), if γ is a gate for ρ other than the first or last, then γ is contained in a line of angle $\pm\delta$, and the gates preceding and following γ do not lie on the same side of this line. Suppose ρ passes through the gate γ at s, and suppose the segment $\rho_{r:s}$ preceding $\rho(s)$ has angle in $(\delta, -\delta)$. Then the gate preceding γ lies east of γ, and the following gate cannot lie east of γ. Consequently the angle of the segment $\rho_{s:t}$ following ρ must lie in $[\delta, -\delta]$.

Next we prove by contradiction that no segment of $\rho_{s:t}$ has angle θ. Suppose $\rho_{x:y}$ is a segment of ρ whose angle is θ. Because ρ is tight in its tunnel, the points $\rho(x)$ and $\rho(y)$ must be endpoints of gates; these gates are collinear and disjoint. Since $\omega_{s:t}$ passes through the same gates, its subpath $\omega_{x:y}$ also has angle θ. But by Lemma 7d.3, the angle of $\omega_{x:y}$ lies in the interval $[\theta^{\top}, -\theta^{\perp}]$ which contains $-\theta$ but not θ.

The clockwise ordering on C can now be broken; it gives rise to a total ordering on the angles of segments of $\rho_{s:t}$. We write $\dot{\alpha}' < \dot{\alpha}$ for $\dot{\alpha}' \in (\theta, \dot{\alpha})$. It remains to prove that the segment α which minimizes $\dot{\alpha}$ satisfies $\dot{\alpha} \ge \theta^{\top}$, and that the segment α which maximizes $\dot{\alpha}$ satisfies $\dot{\alpha} \le -\theta^{\perp}$.

The two cases are essentially alike, so we consider only one. Let α be the segment of ρ that minimizes $\dot{\alpha}$. We may assume $\dot{\alpha} \in (\theta, -\theta)$, and we have $\alpha = \rho_{x:y}$ where $s \le x < y \le t$. The points $\omega(x)$ and $\omega(y)$ lie on the gates containing $\rho(x)$ and

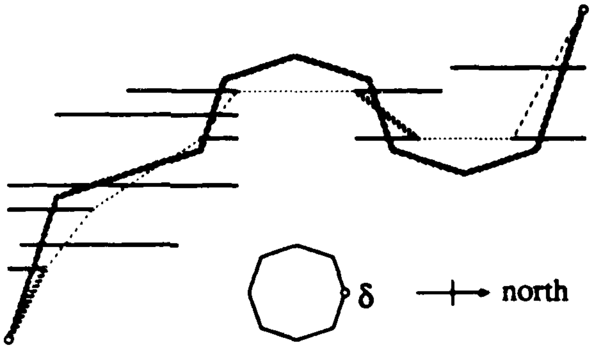


Figure 7e-1. *A restricted route compared to its track.* The shaded path represents the δ-subpath $\omega_{s:t}$ in Lemma 7e.2. The striped and dotted segments make up the corresponding subpath $\rho_{s:t}$ of the restricted δ-route. (The angle δ is shown on its unit polygon.) Dark segments are gates, and the striped segments are the segments of $\rho_{s:t}$ whose angles are "minimal" and "maximal".

$\rho(y)$, respectively. Write $\kappa = \omega_{x:y}$. Lemma 7d.3 implies $\dot{\kappa} \geq \theta^{\top}$. It suffices to show that $\rho(x)$ is at the north end of its gate, and $\rho(y)$ is at the south end of its gate, for then $\dot{\alpha}$ is maximal over all paths between the same gates. In particular, $\dot{\alpha} \geq \dot{\kappa}$ which implies $\dot{\alpha} \geq \theta^{\top}$. Now either $x = s$, in which case the gate corresponding to x is a point, or else $x > s$, when the segment preceding α must have greater (more clockwise) angle (but still within $(\theta, -\theta]$). Hence ρ turns at x, whence by Proposition 7e.1, $\rho(x)$ is an endpoint of its gate. Moreover, since ρ turns to the north at x, the point $\rho(x)$ is the north endpoint of its gate. Entirely symmetrical reasoning proves that $\rho(y)$ is the south endpoint of its gate. □

Among other things, Lemma 7e.2 implies that restricted and partial routes are alike. This fact is a consequence of Proposition 7e.1 and the following lemma.

Lemma 7e.3. *Let ρ be the restricted δ-route of a track ω. If ω has a tie σ at s with $\dot{\sigma} = \pm\delta$, then σ supports ρ at s.*

Proof. Because σ is a tie, ω turns toward $\sigma(0)$ at s. In particular, the angles of the segment of ω following s and the reverse of the segment of ω preceding s lie in some interval $(\phi, -\phi)$ that contains $-\dot{\sigma}$. Because these two segments are tangent to σ, the angles $\dot{\sigma}^{\top}$ and $-\dot{\sigma}^{\perp}$ also lie in $(\phi, -\phi)$. Let $\omega_{s:t}$ be the $\dot{\sigma}$-subpath of ω that begins at s, and let α be the segment of ρ that begins at s. Lemma 7e.2 says that either $\alpha = \omega_{s:t}$ or else $\dot{\alpha} \in [\theta^{\top}, -\theta^{\perp}]$, and Lemma 7d.3 pegs θ at $\dot{\sigma}$. In either case $\dot{\alpha}$ lies in $(\phi, -\phi)$. Now let $\omega_{r:s}$ be the $\dot{\sigma}$-subpath of ω that ends at s, and let α be the segment of ρ that ends at s. Again, Lemma 7e.2 says that either $\alpha = \omega_{r:s}$ or else $\dot{\alpha} \in [\theta^{\top}, -\theta^{\perp}]$. This time σ plays the role of τ in Lemma 7d.3, and hence $\theta = -\dot{\sigma}$. Therefore $-\dot{\alpha} \in [\dot{\sigma}^{\top}, -\dot{\sigma}^{\top}] \subset (\phi, -\phi)$.

Thus the segments α and α' of ρ incident on s have angles in $(\phi, -\phi)$. Moreover, ρ is a shortest path through a δ-tunnel. Hence $\rho(s)$ lies on a gate γ in this tunnel, and the line L through σ separates the gates preceding and following γ in the tunnel. (More precisely, neither open half-plane of L intersects both those gates.) In the same sense, α and α' are also separated by the line. We conclude that $\sigma(0)$ is not exterior to the angle formed by α and α'. In other words, ρ turns toward $\sigma(0)$ at s; the tie σ supports ρ at s. □

Corollary 7e.4. *Restricted and partial routes are alike.*

Proof. Let δ be a diagonal angle; let ω be a tight track with restricted δ-route ρ. By Proposition 7e.1 and the definition of δ-route, it suffices to show that ρ is a tight path through the δ-tunnel for ω. Where the δ-tunnel agrees with the restricted δ-tunnel, this behavior is guaranteed by Proposition 7e.1. Elsewhere it is guaranteed by Lemma 7e.3. For suppose γ is a gate at parameter s in the δ-tunnel for ω, and suppose that γ restricts ω at s; say $\gamma(0) = \omega(s)$. Because ω is tight, it is supported by a tie σ at s, and $\dot\sigma = \dot\gamma$. Now by Lemma 7e.3, the δ-route ρ is also supported by σ at s. Consequently γ restrains ρ at s. □

Retained joints and segments

The precise statement of the correspondence between tracks and their partial routes requires some new definitions. If δ is a diagonal angle, a **δ-rail** of a track ω is a straight δ-subpath of ω. All δ-rails are generically called **rails**. The δ-rails are those we can identify from partial routes. Let s be a joint of a δ-route ρ, let σ be a tie supporting ρ at s, and let $\rho_{r:s}$ and $\rho_{s:t}$ be the segments of ρ just preceding and following s. (Note that ρ must turn at s.) The joint s is **retained** if $\rho_{r:s}$ and $\rho_{s:t}$, when reflected through $\rho(s)$, intersect the polygon $P(\sigma)$ at $\sigma(1)$ only. If ρ is not straight, a segment α of ρ is **retained** if for every path σ that supports it, say at s, the joint s is retained and α does not intersect the inside of $P(\sigma)$.

The definition of retention corresponds directly to the rules given in Section 1D for merging partial realizations. For our purposes here, it is convenient to restate them in terms of angles. If σ supports the partial route ρ at s, the joint s is retained if the segments α and α' preceding and following s, respectively, satisfy $\dot\alpha \notin [\dot\sigma^{\top}, -\dot\sigma^{\perp}]$ and $\dot\alpha' \notin [-\dot\sigma^{\top}, \dot\sigma^{\perp}]$. A segment $\alpha = \rho_{s:t}$ is retained if (1) $\alpha \neq \rho$, and (2) those points of $\{s, t\}$ that are joints of ρ are retained, and (3) every path σ supporting α satisfies $\dot\alpha \in [\dot\sigma^{\perp}, \dot\sigma^{\top}] \cup [-\dot\sigma^{\perp}, -\dot\sigma^{\top}]$.

Proposition 7e.5. *Let ω be a tight track through a maze. The joints of ω are the retained joints of the partial routes of ω. The rails of ω are the retained segments of the partial routes of ω.*

Proof. Let δ be a diagonal angle, and let ρ be the restricted δ-route of ω. By Corollary 7e.4, we may use ρ in place of the δ-route of ω. We show that the joints of ω supported by ties of angles $\pm\delta$ are precisely the retained joints of ρ, and that the δ-rails of ω are precisely the retained segments of ρ.

First consider joints. Let s be a joint of ω corresponding to a tie σ of angle $\pm\delta$; then s is also a joint of ρ, by Lemma 7e.3, and σ supports ρ at s. Let $\omega_{r:s}$ and $\omega_{s:t}$

be the δ-subpaths of ω preceding and following s, respectively; let α and α' be the segments of ρ preceding and following s. To show that s is retained, we must prove $\dot{\alpha} \notin [\dot{\sigma}^{\top}, -\dot{\sigma}^{\perp}]$ and $\dot{\alpha}' \notin [-\dot{\sigma}^{\top}, \dot{\sigma}^{\perp}]$. First apply Lemmas 7e.2 and 7d.3 to $\omega_{r:s}$. There are two cases.

(1) If $\omega_{r:s}$ is straight, it equals α and is tangent to σ. Then we have $\dot{\alpha} \in [\dot{\sigma}^{\perp}, \dot{\sigma}^{\top})$ or $\dot{\alpha} \in (-\dot{\sigma}^{\perp}, -\dot{\sigma}^{\top}]$ according to whether ρ leaves σ to the right or the left. (See the proof of Lemma 7d.2.) In either case $\dot{\alpha} \notin [\dot{\sigma}^{\top}, -\dot{\sigma}^{\perp}]$.

(2) Otherwise each segment of $\rho_{r:s}$, and α in particular, satisfies $\dot{\alpha} \in [\theta^{\top}, -\theta^{\perp}]$ where θ, according to Lemma 7d.3, is $-\dot{\sigma}$. Simplifying, $\dot{\alpha}$ lies in the interval $[-\dot{\sigma}^{\top}, \dot{\sigma}^{\perp}]$, which does not intersect its opposite $[\dot{\sigma}^{\top}, -\dot{\sigma}^{\perp}]$.

The analysis of α' uses the same method.

(1) If $\omega_{s:t}$ is straight, it equals α' and is tangent to σ. Then we have $\dot{\alpha}' \in (\dot{\sigma}^{\perp}, \dot{\sigma}^{\top}]$ or $\dot{\alpha} \in [-\dot{\sigma}^{\perp}, -\dot{\sigma}^{\top})$ according to whether ρ leaves σ to the right or the left. In either case $\dot{\alpha}' \notin [-\dot{\sigma}^{\top}, \dot{\sigma}^{\perp}]$.

(2) Otherwise each segment of $\rho_{s:t}$, and α' in particular, satisfies $\dot{\alpha}' \in [\theta^{\top}, -\theta^{\perp}]$ where θ, according to Lemma 7d.3, is $\dot{\sigma}$. Therefore $\dot{\alpha}'$ does not lie in the opposite interval $[-\dot{\sigma}^{\top}, \dot{\sigma}^{\perp}]$.

We conclude that the joint s is retained.

Now let s be a joint of ρ such that ω is not supported by a tie of angle $\pm\delta$ at s; we show that s is not retained. Let $\omega_{r:t}$ be the δ-subpath of ω with $s \in (r, t)$, and apply Lemma 7e.2. If $\omega_{r:t}$ were straight, it would equal $\rho_{r:t}$, and s could not be a joint of ρ. Therefore the other case applies, so every segment α of $\rho_{s:t}$ satisfies $\dot{\alpha} \in [\theta^{\top}, -\theta^{\perp}]$ where $\theta = \pm\delta$. Now let σ support ρ at s, and let α and α' be the segments of ρ preceding and following s. If $\theta = \dot{\sigma}$, then $\dot{\alpha} \in [\dot{\sigma}^{\top}, -\dot{\sigma}^{\perp}]$, and if $\theta = -\dot{\sigma}$, then $\dot{\alpha}' \in [-\dot{\sigma}^{\top}, \dot{\sigma}^{\perp}]$. Either way s is not retained. We conclude that the retained joints of ρ are exactly the joints of ω supported by ties of angles $\pm\delta$.

Now consider rails. Suppose $\omega_{s:t}$ is a δ-rail of ω. Then $\omega_{s:t}$ is a straight δ-subpath of ω, which implies (Lemma 7e.2) that $\alpha = \rho_{s:t}$ is a segment of ρ and $\alpha \neq \rho$. Furthermore, the endpoints of α that are joints are retained, as we just showed, and α is tangent to at least one tie of slope $\pm\delta$. Therefore $\dot{\alpha} \in [\dot{\sigma}^{\perp}, \dot{\sigma}^{\top}] \cup [-\dot{\sigma}^{\perp}, -\dot{\sigma}^{\top}]$, and α is retained. Now let α be a segment of ρ that is not also a segment of ω. If either supported endpoint is not retained, then α is not retained. Otherwise α connects the endpoints of a δ-subpath of ω. Then Lemma 7d.3 implies $\dot{\alpha} \in (\theta^{\top}, -\theta^{\perp})$ where $\theta = \pm\delta$. Consequently α is not retained, and this observation completes the proof. □

Merging partial routes

Proposition 7e.5 and Lemma 7d.2 imply that a tight track through a maze may be constructed by merging its partial routes. If all the partial routes are straight, then so is the track. Otherwise there is exactly one partial route whose first segment is retained. One constructs the track beginning with this segment, and proceeding to merge the retained vertices of the partial routes according to the following rules.

(1) Suppose the vertex just added was the joint s of the partial route ρ. If ρ has a retained segment $\rho_{s:t}$ beginning at s, then add the other endpoint $\rho(t)$. Stop if $t = 1$, and otherwise repeat.

(2) Choose the partial route whose angles are clockwise or counterclockwise from that of ρ, according to whether the track turns right or left at s. (If σ supports ρ at s, then the track turns right at s if and only if the angle of its preceding segment lies in $(\dot{\sigma}, -\dot{\sigma})$.) Add the first unused retained joint of the new partial route, and return to step 1.

When we run out of retained segments for a given diagonal slope, and hence fall through to step 2, the next segment is supported by ties of different slopes and hence is not retained. Instead, the next vertex is a retained joint of a different partial route, the cho:ce of which is determined by Lemma 7d.2. The correctness of the whole procedure may be proved by a straightforward induction.

Chapter 8

Return to the Sketch Model

At last we return to the model in which the algorithms of Chapter 1 operate. With the knowledge gained in previous chapters, we can now establish the correctness of Algorithm T, which tests the routability of a sketch, and Algorithm R, which produces an optimal routing of a routable sketch.

As the specifications of Algorithms T and R are fairly abstract, so their correctness proofs also avoid formal analysis of low-level details. That is, I am more interested in justifying the ideas behind the algorithms than any particular implementation of those ideas. (I do pay some attention to the algorithms' primary data structure, the rubber-band equivalent, because it accounts for their fast running times.) Most of the ideas behind Algorithms T and R were developed in Chapter 7, so the present chapter is fairly short. Its main concern is building a correspondence between sketches and designs, in order to apply the results of Chapters 6 and 7 to the sketch model.

There are two major differences between the sketch and design models. One is that the terminals of a trace in a proper sketch cannot have overlapping territories, whereas the terminals of a wire in a proper design can have overlapping extents. This discrepancy has already been addresses through the definitions (at the beginning of Chapter 7) of $\sharp$-routability and $\sharp$-safety for designs. The second difference, which is much more profound, is that terminals in a sketch are points, whereas terminals in a design have positive diameter. Since the width of a wire cannot exceed the width of its terminals, the extent of a wire-containing article of a design must have bulges at each end of the wire. In a sketch, however, the territory of a trace may subsume the territories of its terminals. Consequently there are sketches that cannot be adequately represented by any design. Instead we must relate a sketch to a sequence of designs with smaller and smaller fringes. Disregarding certain technicalities, the sketch model is the limit of the design model as fringes collapse to points and line segments.

What follows is a brief outline of this chapter. The connection between sketches and designs is defined in Section 8A and strengthened in Section 8B. These two sections culminate in proofs of the sketch routing and routability theorems (see

Section 1A), which underlie Algorithms R and T. Then in Section 8C we combine these theorems with the results of Chapter 7 to explain the workings of the rubber-band equivalent and, assuming that this data structure does its job and that the scanning procedures are implemented correctly, prove that Algorithms T and R perform as advertised.

8A. The Correspondence Between Designs and Sketches

This section describes a standard method of converting sketches to designs, and thus prepares us to recast results concerning designs in the sketch model. As we show, the correspondence preserves basic properties like homotopy: if one sketch is a realization of another, then the design corresponding to the first sketch will be an embedding of the design corresponding to the second. This section gets as far as showing that the congestion of a cut in a sketch equals the congestion of the corresponding cut in the design. The next section develops the correspondence further.

We relate designs to a subclass of sketches that is restricted in two ways. First, we concentrate on sketches that include a *bounding obstacle*, an island that encloses all the other elements of the sketch. It corresponds to the outer fringe of a design. One can add a bounding obstacle to any sketch; by making it sufficiently large, the routability and routing problems are unaffected. Second, we assume that the routing region of a sketch is connected. Again there is no loss of generality, because a sketch whose routing region is not connected can be analyzed as two or more independent sketches; the wires in different components of the routing region do not interact.

Restricted sketches

For the purposes of this chapter, a sketch is an ordered pair (Ξ, Θ), where Ξ is a finite set of **features** (points and line segments in R^2) and Θ is a finite set of **traces** for Ξ. Let X denote the union $\bigcup_{\xi \in \Xi} \xi$. A trace for Ξ is a simple path in R^2 such that $\theta^{-1}(X) = \{0, 1\}$, and the terminals $\theta(0)$ and $\theta(1)$ of θ are **pointlike** features of Ξ—they are points, and they intersect no other features in Ξ. In addition, the sketch (Ξ, Θ) must satisfy three conditions:

(1) No two members of Ξ intersect other than at their endpoints;

(2) Some of the features in Ξ form a polygon C such that $C \cup \mathit{inside}(C)$ includes X and every trace in Θ; and

(3) The **routing region** $\mathit{inside}(C) - X$ is connected.

The components of X are the **islands** of the sketch (Ξ, Θ); the island containing C is called the **bounding** obstacle. The islands and traces of (Ξ, Θ) are also called the **elements** of that sketch.

For most of the definitions relating to sketches, I refer you to Section 1A. But there are two sets of definitions we should review: those concerning bridges, and those concerning congestion. Formally, a piecewise linear path α in R^2 is a **bridge** for the features Ξ if $\alpha^{-1}(X) = \{0, 1\}$, where X is defined as before. The natural notion of homotopy in the sketch (Ξ, Θ) is that of a **bridge homotopy**, which is a piecewise linear map $F: I \times I \to R^2$ such that $F(\cdot, t)$ is a bridge for Ξ for every $t \in I$. If both $F(0, \cdot)$ and $F(1, \cdot)$ are constant, then we call F a **trace homotopy** also. Two bridges α and β are **bridge-homotopic** (or **trace-homotopic**) if there is a bridge homotopy (or trace homotopy) F such that $F(\cdot, 0) = \alpha$ and $F(\cdot, 1) = \beta$.

In this chapter we consider a **cut** of a sketch to be any bridge in that sketch. The entanglement of a cut and a trace is the minimum number of crossings of that cut by any route for that trace, and the congestion of a cut is a weighted sum of entanglements. The point that needs clarification is the definition of crossing. A **crossing** between two bridges α and β is a point $(s, t) \in I \times I$ such that $\alpha(s) = \beta(t)$ and neither s nor t is 0 or 1. In other words, crossings that occur at endpoints are ignored.

Overview of the correspondence

The correspondence between sketches and designs is parameterized by a positive quantity ϵ that we think of as decreasing to 0. Given a sketch Σ with features Ξ and traces Θ, and given a sufficiently small quantity $\epsilon > 0$, we construct a sheet S_ϵ from Ξ and a set of paths Ω_ϵ from Θ. For sufficiently small ϵ the set Ω_ϵ is a design of wires in the sheet S_ϵ. One can then relate cuts in the sketch Σ to cuts of the sheet S_ϵ, and relate realizations of traces in T to embeddings of wires in the design Ω_ϵ. If X is an object related to Σ, the corresponding object in S_ϵ will be called $\flat_\epsilon(X)$, or simply $X^\flat$, the parameter ϵ being understood. If Y is an object related to S_ϵ, the corresponding object for Σ will be called $\sharp_\epsilon(Y)$ or $Y^\sharp$. As a mnemonic, note that the object $X^\flat$ lies in the *flat* manifold S_ϵ.

$$\text{the sketch } (\Xi, \Theta) \xrightarrow{\flat_\epsilon} \text{the design } \Omega_\epsilon \text{ on } S_\epsilon$$

$$\text{the sketch } (\Xi, \Theta) \xleftarrow{\sharp_\epsilon} \text{the design } \Omega_\epsilon \text{ on } S_\epsilon$$

The operations called $\flat_\epsilon$ and $\sharp_\epsilon$ are not, in general, inverses. The compositions $\sharp_\epsilon \circ \flat_\epsilon$ and $\flat_\epsilon \circ \sharp_\epsilon$ will both be denoted $\natural_\epsilon$; context will determine which order of operations makes sense.

As the parameter ϵ approaches zero, the correspondence between the models becomes tighter. A statement involving ϵ is said to hold **eventually** if it holds for all ϵ less than some positive number δ. For example, Lemma 8a.1 says that if θ is a trace in the sketch Σ, then $\theta^\flat = \flat_\epsilon(\theta)$ is eventually a wire in S_ϵ. If two entities f and g depending on ϵ are eventually equal, we say that f **settles** at g (or vice versa). Proposition 8a.5 says that for any straight cut α in Σ, the congestion of $\alpha^\flat$ settles at the congestion of α. This result is instrumental in achieving the goal of Section 8B, namely, to derive the sketch routability theorem from the design routability theorems of Section 6C.

From sketches to designs

The sheet S_ϵ is constructed as follows. Let B denote the bounding obstacle of the sketch $\Sigma = (\Xi, \Theta)$, and let Z be the set of points inside B that lie at least ϵ units from the features in Ξ, as measured in the wiring norm. Suppose ϵ is less than half the distance between the closest pair of disjoint features. We then define S_ϵ to be Z. Why is this reasonable? The set Z is nonempty, and because the norm $\|\cdot\|$ is polygonal, its boundary $\mathit{Fr}\, Z$ consists of line segments. For each island C other than B, the set $\mathit{Fr}\, Z$ includes a polygon $\flat_\epsilon(C)$ surrounding C whose points lie distance ϵ from C. These polygons are disjoint, and form the inner fringes of S_ϵ. Similarly, $\mathit{Fr}\, Z$ contains a polygon $\flat_\epsilon(B)$ whose points lie distance ϵ from B; this polygon surrounds all the others, and forms the outer fringe of S_ϵ. Each fringe $\flat_\epsilon(C)$ of S_ϵ is considered to have the same width as the island C of Σ.

Next we define $\flat_\epsilon(\theta)$ for a bridge θ in Θ. Let A and B denote the fringes of S_ϵ that correspond to the terminals of θ. Then A surrounds $\theta(0)$ and B surrounds $\theta(1)$. Let s be the point at which θ leaves A, and let t be the point at which w enters B. In symbols, we define $s = \sup \theta^{-1}(A)$ and $t = \inf \theta^{-1}(B)$. We have $\theta(s) \in A$ and $\theta(t) \in B$ because A and B are closed sets. The path $\theta^\flat$ is just $\theta_{s:t}$. If Γ is any set of bridges in Σ, we put $\Gamma^\flat = \{\, \flat_\epsilon(\gamma) : \gamma \in \Gamma \,\}$. The design Ω_ϵ is simply $\Theta^\flat = \{\, \theta^\flat : \theta \in \Theta \,\}$. Each path $\theta^\flat$ in Ω_ϵ is assigned the same width as the corresponding trace θ.

The paths $\theta^\flat$ in Ω_ϵ are not always wires, but they are wires if ϵ is small enough. Our first lemma implies that Ω_ϵ is eventually a design, and that a realization of the sketch Σ eventually corresponds to an embedding of Ω_ϵ.

Lemma 8a.1. *Let (Ξ, Θ) be a sketch. If β is a bridge for Ξ, then $\beta^\flat$ is eventually a link in S_ϵ. And if β and θ are bridge-homotopic with respect to the features Ξ, then $\beta^\flat$ and $\theta^\flat$ are eventually link-homotopic.*

Proof. Let X represent the set of points contained in the features Ξ. For any subset C of R^2, the notation $\|C - X\|$ denotes the distance between C and X: the infimum of $\|c - x\|$ over all $c \in C$ and all $x \in X$. We say a path α in R^2 **flees** from Ξ if the function $t \mapsto \|\alpha(t) - X\|$ is increasing. If β is any bridge for Ξ, then there are points $s, t \in I$ satisfying $0 < s < t < 1$ such that $\beta_{0:s}$ and $\beta_{1:t}$ flee from Ξ.

(Since β is a bridge, for sufficiently small s and sufficiently large t the paths $\beta_{0:s}$ and $\beta_{1:t}$ are straight.) Let s and t be chosen thus.

I claim that if $\epsilon < \|Im\,\beta_{s:t} - X\|$, then $\beta^\flat$ is a link in S_ϵ. First of all, $\beta_{s:t}$ is a path in S_ϵ by the choice of ϵ. Second, the paths $\beta_{0:s}$ and $\beta_{1:t}$ can intersect $Bd\,S_\epsilon$ in at most one point, since they flee from Ξ. If P and Q are the terminals of β, it follows that $\beta^\flat$ is a link from $P^\flat$ to $Q^\flat$.

The argument carries over to homotopies nearly intact. Let H be a bridge homotopy between β and θ. That means $\beta_x = H(\cdot, x)$ is a bridge for all $x \in I$, and H itself is piecewise linear. We find a parameter $s > 0$ such that $(\beta_x)_{0:s}$ flees from Ξ for every $x \in I$. First choose s small enough so that for every x, the path $(\beta_x)_{0:s}$ has at most two segments. This is possible because H is piecewise linear. For each point $x \in I$ choose $s_x > 0$ so that $(\beta_x)_{0:s_x}$ flees from Ξ. Using the continuity of H one can find a neighborhood I_x of x in I such that $(\beta_y)_{0:s_x}$ flees from Ξ for each point $y \in I_x$. Because I is compact, finitely many of these intervals I_x cover I; let s be the minimum of the corresponding values of s_x. By a symmetrical argument there exists $t < 1$ such that $(\beta_x)_{1:t}$ flees from Ξ for all $x \in I$. If necessary, decrease s or increase t so that $s < t$. The compact set $C = H([s,t], I)$ does not intersect any feature in Ξ, so the quantity $\|C - X\|$ is positive.

I show that if ϵ is less than $\|C - X\|$, then $\beta^\flat$ and $\theta^\flat$ are link-homotopic links in S_ϵ. Because $\|C - X\| \leq \|Im(\beta_x)_{s:t} - X\|$ for any x, the claim implies that $(\beta_x)^\flat$ is a link in S_ϵ. Hence the map $u, x \mapsto (\beta_x)^\flat(u)$ is a link homotopy provided it is continuous. By the definition of $(\beta_x)^\flat$, it suffices to show that the functions $f\colon x \mapsto \sup(\beta_x)^{-1}(P^\flat)$ and $g\colon x \mapsto \inf(\beta_x)^{-1}(Q^\flat)$ are continuous. We consider only the first. Because $(\beta_x)_{0:s}$ flees from Ξ, the set $\sup(\beta_x)^{-1}(P^\flat)$ has a unique member u_x. The graph of the function $f\colon x \mapsto u_x$ is precisely the set $H^{-1}(P^\flat)$, which is closed because H is continuous. A map into a compact Hausdorff space is continuous if and only if its graph is closed. Therefore f is a continuous function. □

If the input β to Lemma 8a.1 is a trace, its output $\beta^\flat$ is a wire. Suppose β is a trace for the features Ξ, and let P and Q be its terminals. Eventually $\beta^\flat$ is a link in S_ϵ. Because P and Q are pointlike, the polygons $P^\flat$ and $Q^\flat$ are convex inner fringes of S_ϵ. Because β is simple, so is $\beta^\flat$. Therefore $\beta^\flat$ is eventually a wire in S_ϵ.

Corollary 8a.2. *If (Ξ, Θ) is a sketch, then eventually $\Theta^\flat$ is a design on S_ϵ. If (Ξ, Φ) is a realization of (Ξ, Θ), then eventually $\Phi^\flat$ is an embedding of $\Theta^\flat$.*

Proof. For each trace $\theta \in \Theta$, eventually $\theta^\flat$ is a wire in S_ϵ. Since Θ is finite, eventually the set $\Theta^\flat$ contains only wires. That set is a design. Its wires are disjoint, because they are subpaths of the disjoint traces in Θ, and no two share a terminal, because no two traces in Θ share a terminal. Now bring Φ into the picture. For each trace $\theta \in \Theta$ there exists a trace $\phi \in \Phi$ that is bridge-homotopic to θ. Eventually $\Phi^\flat$ is a design, and by Lemma 8a.1, eventually $\theta^\flat$ is link-homotopic to $\phi^\flat$ whenever θ is bridge-homotopic to ϕ. When this occurs, $\Phi^\flat$ is an embedding of $\Theta^\flat$. □

From designs to sketches

Now we show how to convert wires in S_ϵ into traces for the features Ξ. Let X be the union of the features in Ξ. First define a piecewise linear map $\sharp_\epsilon\colon Bd\, S_\epsilon \to X$ that sends each point p on a fringe $C^\flat$ to a point $p^\sharp$ on the island C. Choose this function so that for any two distinct points p and q on the fringe $C^\flat$, the straight paths $p \triangleright p^\sharp$ and $q \triangleright q^\sharp$ touch neither X nor each other except at $p^\sharp$ and $q^\sharp$. Now define $\sharp_\epsilon(\omega)$ for a wire ω in S_ϵ by

$$\sharp_\epsilon(\omega) = \big(\omega(0)^\sharp \triangleright \omega(0)\big) \star \omega \star \big(\omega(1) \triangleright \omega(1)^\sharp\big).$$

With this definition, we obtain counterparts to Lemma 8a.1 and Corollary 8a.2. They imply that an embedding of the design Ω_ϵ always corresponds to a realization of the sketch Σ.

Lemma 8a.3. *Let (Ξ, Θ) be a sketch. If γ is a piecewise linear link in a sheet S_ϵ, then $\gamma^\sharp$ is a bridge for Ξ. If γ is link-homotopic to a link χ, then $\gamma^\sharp$ is bridge-homotopic to $\chi^\sharp$.*

Proof. That $\gamma^\sharp$ is a bridge is clear. Let F be a link homotopy between γ and χ. By Lemma 2c.7, we may assume that F is piecewise linear. Define a function $G\colon I \times I \to R^2$ by $G(\cdot, x) = \sharp_\epsilon(F(\cdot, x))$; then $G(\cdot, 0) = \gamma^\sharp$ and $G(\cdot, 1) = \chi^\sharp$. I claim that G is a bridge homotopy. First of all, G is continuous because $\sharp_\epsilon(\eta)$ is a continuous function of η. Second, G is piecewise linear because F and $\sharp_\epsilon$ are. Third, for each $x \in I$, the middle of $G(\cdot, x)$ intersects no feature in Ξ. Finally, the sets $F(0, I)$ and $F(1, I)$ are subsets of fringes $P^\flat$ and $Q^\flat$, where P and Q are terminals of Ξ; hence $G(0, I)$ and $G(1, I)$ are subsets of P and Q, respectively. Thus G is a bridge homotopy with respect to the features Ξ. □

If the input γ to Lemma 8a.3 is a wire whose terminals correspond to pointlike features, then the output $\gamma^\sharp$ is a trace. Let ω be a wire in S_ϵ with terminals $P^\flat$ and $Q^\flat$, where P and Q are pointlike features of Ξ. Because ω is piecewise linear, so is $\omega^\sharp$; because ω is simple and its endpoints lie on different fringes, $\omega^\sharp$ is simple. Furthermore $\gamma^\sharp$ intersects the features of Ξ only at its terminal points P and Q. Therefore $\gamma^\sharp$ is a trace for Ξ.

Correspondence of congestion

Next we show that corresponding cuts eventually have equal congestion. In addition to Lemmas 8a.1 and 8a.3, we need one further fact. The following lemma shows that $\flat_\epsilon$ and $\sharp_\epsilon$ are "nearly" inverses, at least with regard to bridge homotopy and entanglement.

Lemma 8a.4. *If α is a bridge in the sketch (Ξ, Θ), then eventually $\alpha^\natural$ is bridge-homotopic to α and $tangle(\alpha^\natural, \theta) = tangle(\alpha, \theta)$ for all traces $\theta \in \Theta$.*

Proof. Put $X = \bigcup_{\xi \in \Xi} \xi$, and let R be the union of X with the routing region. The path $\alpha^\natural = \sharp_\epsilon \circ \flat_\epsilon(\alpha)$ has the form

$$\alpha^\natural = \big(\alpha(s)^\sharp \triangleright \alpha(s)\big) \star \alpha_{s:t} \star \big(\alpha(t) \triangleright \alpha(t)^\sharp\big).$$

Let β denote the path $\alpha_{0:s} \star \alpha_{s:t} \star \alpha_{t:1}$, which is a reparameterization of α. For sufficiently small ϵ the paths $\alpha_{0:s}$ and $\alpha_{t:1}$ are linear, and then β differs from $\alpha^\natural$ only in its first and last segments. One can construct a piecewise linear motion of the plane, constant except near the endpoints of α, that takes β onto $\alpha^\natural$. In other words, there is a piecewise linear map $F: R \times I \to R$ such that (a) $F(\,\cdot\,, 0) = id_R$, (b) $F(X, I) = X$, (c) $F(\,\cdot\,, t)$ is a homeomorphism of $R - X$ with itself for each $t \in I$, and (d) $F(\,\cdot\,, 1) \circ \beta = \alpha^\natural$. The map $s, t \mapsto F(\alpha(s), t)$ is a bridge homotopy between β and $\alpha^\natural$, and consequently α and $\alpha^\natural$ are bridge-homotopic.

Now let θ be any trace in Θ; we show that $tangle(\alpha^\natural, \theta) = tangle(\beta, \theta)$. The lemma will follow, since β is just a reparameterization of α. Some route η_0 for θ makes only $n = tangle(\beta, \theta)$ crossings with β. We find a route η_1 for θ that makes only n crossings with $\alpha^\natural$. For $t \in I$, define $\eta_t = F(\,\cdot\,, t) \circ \eta$; by property (a) above, this definition is consistent with η_0. Properties (b) and (c) above imply that $\eta_t^{-1}(X) = \{0, 1\}$ for each t. And since F is piecewise linear, so is η_t, and so is the homotopy $G: s, t \mapsto \eta_t(s)$ between η_0 and η_1. Thus η_1 is a bridge, and G is a bridge homotopy. Therefore η_1 is a route for η_0, and hence of θ. Finally, because $\alpha^\natural = F(\,\cdot\,, 1) \circ \beta$ (property (d)) and $\eta_1 = F(\,\cdot\,, 1) \circ \eta_0$, and $F(\,\cdot\,, 1)$ is a homeomorphism of $R - X$ with itself, the number of crossings between $\alpha^\natural$ and η_1 (as bridges) is equal to the number of crossings between β and η_0, namely n. Therefore $tangle(\alpha^\natural, \theta) \leq n = tangle(\beta, \theta)$. The opposite inequality is proved similarly. □

For simplicity, our congestion result considers only straight cuts, since those are the only cuts we really need.

Proposition 8a.5. *If α is a straight cut of the sketch (Ξ, Θ), then $cong(\alpha^\flat, \Omega_\epsilon)$ settles at $cong(\alpha)$.*

Proof. Assume ϵ is small enough that Ω_ϵ is a design on the sheet S_ϵ. We show that the entanglement of a trace θ with α is eventually equal to the entanglement of $\theta^\flat$ with $\alpha^\flat$. First, suppose $tangle(\alpha, \theta) = m$. Then θ has a route β that makes only m crossings with α. Hence for any $\epsilon > 0$, the path $\beta^\flat$ makes at most m crossings with $\alpha^\flat$. By Lemma 8a.1, $\beta^\flat$ is eventually link-homotopic to $\theta^\flat$. Therefore $tangle(\alpha^\flat, \theta^\flat)$ is eventually at most m.

Now we show the reverse inequality. Suppose $tangle(\alpha^\flat, \theta^\flat) = n$. Then there is a route ω of $\theta^\flat$ that makes exactly n crossings with $\alpha^\flat$. By Lemma 8a.3, the paths $\omega^\sharp$

and $\theta^\natural$ are bridge-homotopic, whence $\omega^\natural$ is bridge-homotopic to θ by Lemma 8a.4. The endpoints of ω do not lie on $Im\,\alpha^\flat$, or else these crossings could be removed. Hence $\omega^\natural$ and $(\alpha^\flat)^\natural$ cross only where ω and $\alpha^\flat$ do. Because $\omega^\natural$ is a route for θ, we have $tangle(\alpha^\natural, \theta) \leq n$. Lemma 8a.4 now shows that $tangle(\alpha, \theta) \leq n$.

To prove the lemma, let ϵ be small enough that $tangle(\alpha, \theta) = tangle(\alpha^\flat, \theta^\flat)$ for every trace θ in Θ. Then we have

$$\begin{aligned} cong(\alpha) &= \sum_{\theta\in\Theta} width(\theta) \cdot tangle(\alpha, \theta) \\ &= \sum_{\theta\in\Theta} width(\theta^\flat) \cdot tangle(\alpha^\flat, \theta^\flat) \\ &= cong(\alpha^\flat, \Theta^\flat). \end{aligned}$$

Thus $cong(\alpha)$ is eventually equal to $cong(\alpha^\flat, \Omega_\epsilon)$. □

Proposition 8a.5 and Lemma 8a.1 let us carry over our first result from designs to sketches: the invariance of congestion under homotopy of cuts.

Corollary 8a.6. *Bridge-homotopic simple cuts have equal congestion.*

Proof. Let α and β be bridge-homotopic alpha cuts in a sketch $\Sigma = (\Xi, \Theta)$. According to Lemma 8a.1, $\alpha^\flat$ and $\beta^\flat$ are eventually link-homotopic, whence by Proposition 4b.3 they eventually have equal flow. The cuts $\alpha^\flat$ and $\beta^\flat$ are also simple, and hence have congestion equal to their flow, by Proposition 4b.6. Finally, Proposition 8a.5 says that the congestion of α is the value at which $cong(\alpha^\flat, \Theta^\flat)$ settles, and similarly for β. We conclude that these values are equal. □

8B. Sketch Theorems

This section extends the correspondence between sketches and designs to include safety and routability. Specifically, we show that a sketch is safe if and only if the corresponding designs are safe; and a sketch is routable if and only if the corresponding designs are $\natural$-routable. In the process we identify certain *critical* cuts that dominate the others, in the sense that if any cut in a sketch is unsafe and nonempty, then one of the critical cuts in unsafe and nonempty. One product is a strong form of the sketch routability theorem: a sketch is routable if and only if its nonempty critical cuts are safe. Another result is the sketch routing theorem: every trace in a routable sketch has an *ideal* realization that is no longer than any feasible realization of that trace, and the ideal realizations of the traces in a sketch form a proper realization of the whole sketch.

Territories and capacities

Going from a sketch to a design, the islands expand into fringes while retaining their assigned widths. It appears that a sketch on the brink of unroutability would necessarily give rise to unroutable designs. Something must give, and what gives is the wiring norm. Let Σ be a sketch with wiring norm $\|\cdot\|$, and let μ be half the width of the narrowest element of Σ. We measure extents and capacities in S_ϵ with the norm $\|\cdot\|_\epsilon = \frac{\mu}{\mu-\epsilon}\|\cdot\|$. With this definition, the territory of each element C of Σ contains the extent of the corresponding detail $C^\flat$ in S_ϵ. Conversely, every point in the territory of C eventually lies in the extent of $C^\flat$. Together with Lemma 8a.1, these facts allow us to relate the routability of Σ to the $\sharp$-routability of Ω_ϵ.

Lemma 8b.1. *If the sketch $\Sigma = (\Xi, \Theta)$ is safe and $\Theta^\flat$ is a design, then $\Theta^\flat$ is $\sharp$-safe. If Σ is routable, then $\Theta^\flat$ is eventually $\sharp$-routable.*

Proof. First we look at safety. Suppose that Σ is safe and that $\Theta^\flat$ is a design on the sheet S_ϵ. The extent of each detail of $\Theta^\flat$ is contained within that of the corresponding element of (Ξ, Θ). Since the sketch (Ξ, Θ) is safe, none of its elements have overlapping territories, except where the territories of wires overlap with the territories of their terminals. Hence the same is true of $\Theta^\flat$. Furthermore, the traces in Θ are self-avoiding. If θ is any trace in Θ, then the territory of θ, together with those of its terminals, does not separate any two features in Ξ. Since the extent of the corresponding details of $\Theta^\flat$ are smaller, we conclude that the wires of $\Theta^\flat$ are self-avoiding also. Thus $\Theta^\flat$ is $\sharp$-safe.

Now we look at routability. Corollary 8a.2 says that every realization of (Ξ, Θ) eventually gives rise to an embedding of $\Theta^\flat$. If Σ is routable, it has a safe embedding, which eventually engenders a $\sharp$-safe embedding of $\Theta^\flat$. Thus $\Theta^\flat$ is eventually $\sharp$-routable. □

The next lemma relates the capacities and safety of straight cuts. We say that a cut α in the sketch Σ is **exposed** if $\|\alpha^\flat\|$ eventually equals $\|\alpha\| - 2\epsilon$.

Lemma 8b.2. *Let α be a straight cut in the sketch (Ξ, Θ). If α is unsafe, then $\alpha^\flat$ is eventually unsafe. If α is safe and exposed, then $\alpha^\flat$ is eventually safe.*

Proof. By Proposition 8a.5, we may assume ϵ is so small that the congestion of $\alpha^\flat$ has settled at the value $cong(\alpha)$, which we denote by c. If α is unsafe, then c exceeds the capacity of α by some positive amount. Clearly the capacity of $\alpha^\flat$ converges to that of α as $\epsilon \to 0$, so eventually the congestion of $\alpha^\flat$, which is also c, exceeds $cap(\alpha^\flat)$. Now suppose that α is safe. Then we must have $cap(\alpha) \geq 0$, whence $\|\alpha\| \geq 2\mu$. For the other direction, if eventually $\|\alpha^\flat\| = \|\alpha\| - 2\epsilon$, then $cap(\alpha^\flat)$

eventually differs from *cap*(α) by

$$\begin{aligned}\|\alpha^\flat\|_\epsilon - \|\alpha\| &= \tfrac{\mu}{\mu-\epsilon}(\|\alpha\| - 2\epsilon) - \|\alpha\| \\ &= \tfrac{\mu}{\mu-\epsilon}((\epsilon/\mu)\,\|\alpha\| - 2\epsilon) \\ &\geq \tfrac{\mu}{\mu-\epsilon}(2\epsilon - 2\epsilon) = 0.\end{aligned}$$

Therefore the capacity of $\alpha^\flat$ is eventually no less than that of α, and so $\alpha^\flat$ is eventually safe if α is safe. □

Now we have enough machinery to prove one direction of the sketch routability theorem.

Proposition 8b.3. *A sketch that contains an unsafe, nonempty, straight cut is unroutable.*

Proof. Let α be a nonempty, unsafe, straight cut in the sketch (Ξ, Θ). Then $\alpha^\flat$ is straight (by definition), eventually nonempty (by Proposition 8a.5), and eventually unsafe (by Lemma 8b.2). In other words, for all ϵ less than some positive ϵ_0, the design Ω_ϵ contains a nonempty, unsafe, straight cut. According to Section 6C (see the third line of Table 6c-1), no embedding of Ω_ϵ is $\natural$-proper. By Lemma 8b.1, therefore, the sketch (Ξ, Θ) is unroutable. □

Critical cuts

We would like to prove the converse of Proposition 8b.3: that the design Ω_ϵ is eventually $\natural$-safe if the sketch Σ is safe. This statement is true, but is not an easy consequence of Lemma 8b.2. First of all, not every straight cut α in Σ is exposed. Moreover, even if $\alpha^\flat$ is eventually safe for each straight cut α, it need not happen that eventually all such cuts become safe, since there are infinitely many straight cuts in Σ. To overcome these problems we need a finite set of straight, exposed cuts Γ such that the set $\Gamma^\flat = \{\,\gamma^\flat : \gamma \in \Gamma\,\}$ is eventually $\natural$-*decisive* (Definition 6d.1) for the sheet S_ϵ.

Fortunately, such a cut set is at hand: we let Γ contain the exposed critical cuts in the sketch $\Sigma = (\Xi, \Theta)$. Proposition 6d.8 allows us to show that $\Gamma^\flat$ is eventually a $\natural$-decisive set of cuts in the sheet S_ϵ. Recall from Section 1C that a critical cut is a straight cut that begins at a feature endpoint and travels to the closest point on a disjoint feature, as measured in the wiring norm, with ties broken using the euclidean norm. (Actually, the ties may be broken arbitrarily.) The critical cuts for a sketch depend only on the features and the wiring norm. Consequently the set $\Gamma^\flat$ is independent of the design $\Theta^\flat$.

Proposition 8b.4. *If Γ is the set of exposed critical cuts in the sketch Σ, then $\Gamma^\flat$ is eventually $\natural$-decisive.*

Proof. Let ϵ be small enough that S_ϵ is a sheet, every path in $\Gamma^\flat$ is a link in S_ϵ, and every path $\gamma \in \Gamma$ satisfies $\|\gamma^\flat\| = \|\gamma\| - 2\epsilon$. Once this equation holds, it holds for all smaller values of ϵ, and hence if a cut χ of Σ is critical but unexposed, then $\|\chi^\flat\| < \|\chi\| - 2\epsilon$.

By Proposition 6d.8 and Corollary 6d.4, it suffices to show that $\Gamma^\flat$ spans the sheet S_ϵ. For each feature P of Σ, let $P^\flat$ be the set of points of distance ϵ from P. Then $P^\flat$ is a convex polygon, and the collection of such polygons over all features P of Σ is an edging for S_ϵ. (See Definition 6d.7.) Let $P^\flat$ and $Q^\flat$ be elements of this edging. We must show that either

(1) $\Gamma^\flat$ contains a minimal path from $P^\flat$ to $Q^\flat$ that is a cut in S_ϵ, or

(2) there is a minimal path from $P^\flat$ to $Q^\flat$ that is not a cut in S_ϵ.

We can assume that $P^\flat$ and $Q^\flat$ do not intersect, else case (2) would obtain.

In both cases the minimal path is derived from something like a critical cut. Let χ be a minimal path from P to Q. We may choose χ so that if χ is a cut, either χ or $\widehat{\chi}$ is critical. Let χ^* be the subpath of χ that runs from $P^\flat$ to $Q^\flat$. Its length is $\|P - Q\| - 2\epsilon$ which equals $\|P^\flat - Q^\flat\|$, and hence χ^* is a minimal path from $P^\flat$ to $Q^\flat$. If χ is not a cut, neither is χ^*, and case (2) holds. Assume therefore that χ is a critical cut. If χ is not exposed, then $\|\chi^\flat\| < \|\chi\| - 2\epsilon$ and hence $\chi^\flat \neq \chi^*$. Consequently χ^* is not a cut, and again case (2) occurs. Assume therefore that χ is exposed. Then $\chi \in \Gamma$. Also $\|\chi^\flat\| == \|\chi\| - 2\epsilon$, which means $\chi^\flat = \chi^*$, which leads to case (1). □

Ideal realizations

Just as routable designs have ideal embeddings, routable sketches have ideal realizations. If θ is a trace in a routable sketch (Ξ, Θ), a realization ρ of θ is **ideal** if once the design $\Theta^\flat$ becomes $\sharp$-routable (Lemma 8b.1), the ideal embedding of $\theta^\flat$ converges uniformly to ρ.

Proposition 8b.5. *Let (Ξ, Θ) be a sketch. If the design $\Theta^\flat$ is eventually $\sharp$-routable, then every trace in Θ has a unique ideal realization.* □

Proposition 8b.5 is difficult, and I have not written out a formal proof. I discuss the proof, however, in Section 8C. The only remaining step is the following.

Proposition 8b.6. *If every trace in a sketch has an ideal realization, then those realizations form a proper sketch.*

Proof. Let $\Sigma = (\Xi, \Theta)$ be a sketch, and for each trace θ in Θ, let θ' be the ideal realization of θ. Denote by Θ' the set $\{\theta' : \theta \in \Theta\}$. We show that (Ξ, Θ') is a proper realization of (Ξ, Θ). Three things must be shown:

(1) that no two islands in Σ have overlapping territories;

(2) that no trace in Θ' has a territory that intersects the territory of any other trace in Θ', or the territory of any island in Σ except its terminals; and

(3) that each trace in Θ' is self-avoiding.

Claim (1) is easy. Whenever Ω_ϵ is $\sharp$-routable, no two fringes of S_ϵ have overlapping extents in the norm $\|\cdot\|_\epsilon = \frac{\mu}{\mu-\epsilon}\|\cdot\|$. Hence if P and Q are any two islands in Σ, and $P^\flat$ and $Q^\flat$ are the corresponding fringes of S_ϵ, we eventually have

$$\begin{aligned}\|P - Q\| &\geq \tfrac{\mu-\epsilon}{\mu}\left\|P^\flat - Q^\flat\right\|_\epsilon \\ &\geq \tfrac{\mu-\epsilon}{\mu}\big(\mathit{width}(P) + \mathit{width}(Q)\big)/2.\end{aligned}$$

Since this inequality holds for arbitrarily small ϵ, the distance between P and Q is at least the mean of their widths. Therefore the territories of P and Q do not overlap.

Claim (2) is a little harder, but only because it involves the convergence of ideal embeddings. Let θ be a trace in Θ and P an island of Σ other than the terminals of θ. For ϵ small enough that Ω_ϵ is $\sharp$-routable, let ρ_ϵ denote the ideal embedding of the wire $\theta^\flat \in \Omega_\epsilon$. Let $\delta > 0$ be arbitrary. Because ρ_ϵ converges to θ' uniformly as $\epsilon \rightarrow 0$, eventually $\|\rho_\epsilon(t) - \theta'(t)\| < \delta$ for all $t \in I$. When Ω_ϵ is $\sharp$-routable, the extents of ρ_ϵ and $P^\flat$ do not overlap, so eventually

$$\begin{aligned}\|P - \mathit{Im}\,\theta'\| &> \left\|P^\flat - \mathit{Im}\,\rho_\epsilon\right\| - \delta \\ &= \tfrac{\mu-\epsilon}{\mu}\left\|P^\flat - \mathit{Im}\,\rho_\epsilon\right\|_\epsilon - \delta \\ &\geq \tfrac{\mu-\epsilon}{\mu}\big(\mathit{width}(P) + \mathit{width}(\theta)\big)/2 - \delta.\end{aligned}$$

Since this inequality holds for arbitrarily small δ and ϵ, it holds with $\delta = \epsilon = 0$. Thus the distance from P to θ' is at least the mean of their widths, which implies that their territories are disjoint. A similar argument shows that no two traces in Θ' have overlapping territories.

Claim (3) says that the traces in Θ' are self-avoiding. Suppose to the contrary that $\theta' \in \Theta'$ is not self-avoiding. Then the territories of θ' separates two islands P and Q of Ξ, so by Lemma 2c.2 there is a loop λ within the territory of θ that separates one from the other. Because $\mathit{Im}\,\lambda$ is compact, there is some $\delta > 0$ such that every point of $\mathit{Im}\,\lambda$ lies within $\mathit{width}(\theta)/2 - \delta$ units of $\mathit{Im}\,\theta'$. In other words, for every $s \in I$ there exists $t \in I$ such that $\|\lambda(s) - \theta'(t)\| \leq \mathit{width}(\theta)/2 - \delta$. Since $\rho_\epsilon \rightarrow \theta'$ uniformly as $\epsilon \rightarrow 0$, eventually we have $\|\rho_\epsilon(t) - \theta'(t)\| < \delta/2$ for all $t \in I$. Then by the triangle inequality, for each s there exists t such that $\|\lambda(s) - \rho_\epsilon(t)\| < \mathit{width}(\theta)/2 - \delta/2$. If ϵ is small enough that $(\epsilon/2\mu)\,\mathit{width}(\theta) < \delta/2$, then $\|\lambda(s) - \rho_\epsilon(t)\|_\epsilon < \mathit{width}(\theta)/2$. Now $\mathit{width}(\theta) = \mathit{width}(\rho_\epsilon)$, so this means every point of $\mathit{Im}\,\lambda$ lies within the extent of ρ_ϵ. Therefore the extent of ρ_ϵ eventually separates P from Q, ‹r $P^\flat$ from $Q^\flat$.

(Once Ω_ϵ becomes $\sharp$-routable, the extent of ρ_ϵ cannot intersect either $P^\flat$ or $Q^\flat$.) Therefore ρ_ϵ eventually fails to be self-avoiding, a contradiction. This observation completes the proof. □

The sketch routability and routing theorems

Now we put the pieces together.

Theorem 8b.7. (Sketch Routability Theorem) *A sketch is routable if and only if its nonempty critical cuts are safe.*

Proof. Since critical cuts are straight, Proposition 8b.3 takes care of the "only if" direction. For the "if" direction, suppose Σ is a sketch whose nonempty critical cuts are safe. By Proposition 8b.4, there is a finite set Γ of exposed cuts in Σ such that $\Gamma^\flat$ is eventually $\sharp$-decisive. By Proposition 8a.5, eventually $\gamma^\flat$ is empty in Ω_ϵ whenever γ is empty. And by Lemma 8b.2, $\gamma^\flat$ is eventually safe in Ω_ϵ if γ is safe in Σ, for each $\gamma \in \Gamma$. Since Γ is finite, all the nonempty cuts in $\Gamma^\flat$ are eventually safe, and since $\Gamma^\flat$ is eventually $\sharp$-decisive, this means the design Ω_ϵ is eventually $\sharp$-routable. By Propositions 8b.5 and 8b.6, therefore, Σ has a proper realization. Thus Σ is routable. □

Theorem 8b.8. (Sketch Routing Theorem) *The ideal realizations of the wires in a safe sketch form a proper sketch. They have minimal euclidean arc length among all feasible realizations of those wires.*

Proof. The first statement is merely an elaboration of what we just showed in Theorem 8b.7. Now let ρ be the ideal realization of a trace θ in the safe sketch (Ξ, Θ), and let ρ_ϵ denote the ideal embedding of $\theta^\flat$. Lemma 8b.1 shows that the design corresponding to a proper realization of a sketch (Ξ, Θ) is a proper embedding of the design $\Theta^\flat$. So if β is a feasible realization of a trace $\theta \in \Theta$, then $\beta_\epsilon = \flat_\epsilon(\beta)$ is a feasible embedding of $\theta^\flat$. Now $|\beta_\epsilon|$ converges to $|\beta|$ as $\epsilon \to 0$, and by the design routing theorem (6c.2), the arc length of β_ϵ is at least that of ρ_ϵ. If $|\rho_\epsilon|$ converged likewise to $|\rho|$, then we would have $|\beta| \geq |\rho|$ as desired.

Actually, it suffices to find a lower bound on $|\rho_\epsilon|$ that converges to $|\rho|$. Suppose the joints of ρ are $r_1, \ldots, r_n$, and let γ_ϵ denote the polygonal approximation to ρ_ϵ whose vertices lie at

$$\rho_\epsilon(0), \rho_\epsilon(r_1), \ldots, \rho_\epsilon(r_n), \rho_\epsilon(1).$$

We have $|\gamma_\epsilon| \leq |\rho_\epsilon|$ by the definition of arc length, and because ρ_ϵ converges uniformly to ρ, the arc length $|\gamma_\epsilon|$ converges to $|\rho|$. Since $|\beta_\epsilon| \geq |\rho_\epsilon| \geq |\gamma_\epsilon|$, and $|\beta_\epsilon| \to |\beta|$, we have $|\beta| \geq |\rho|$. Thus ideal realizations have minimal euclidean arc length among all feasible realizations. □

8C. Correctness of the Sketch Algorithms

The time has come to reconsider the algorithms of Chapter 1. Sad to say, this thesis does not prove those algorithms correct. It does, however, show how one could make the connection between the sketch algorithms and the theorems of Chapters 6 and 7 concerning designs. In most cases what is needed is a careful analysis of the design corresponding to a sketch, or more specifically, the way in which something associated with that design tends to a limit as $\epsilon \to 0$. Three good examples come to mind: the ideal embeddings of the wires, the mazes for those wires (discussed below), and the elastic-chain equivalent of the design. These limiting processes are very tedious to evaluate, and I have not worked them all out. I have little doubt, however, that they can be worked out.

In this section I first argue that the rubber-band equivalent of a sketch, in combination with the scanning methods of Algorithms T and R, correctly computes the congestions of straight cuts (in Algorithm T) and the diagonal gates for traces (in Algorithm R). Here I appeal to the results of Section 7C concerning elastic chains. Assuming that the scanning procedures do their jobs, I then argue that Algorithm T checks the safety and emptiness of every critical cut (which is clear), and that Algorithm R produces an ideal realization of any routable input sketch. In discussing Algorithm R I outline the reasons why every trace in a routable sketch has a unique ideal realization, and thereby provide some justification for Proposition 8b.5.

The sketch algorithms are best understood not in terms of sketches, where our mathematical understanding is poor, but rather in terms of the limiting behavior of the corresponding designs. A good example is the rubber-band equivalent of a sketch $\Sigma = (\Xi, \Theta)$. What the RBE of a sketch Σ computes, given a straight cut α, is by definition the content of α: the sequence of rubber bands $\langle \rho_1, \ldots, \rho_n \rangle$ of traces in Σ that necessarily cross α. Say ρ_i is the rubber band of $\theta_i \in Theta$ for each i. We do not interpret this sequence in terms of necessary crossings of α by traces in Σ; no such concept has been defined. Instead we interpret it as a sequence $\langle \rho_1, \ldots, \rho_n \rangle$ such that for all sufficiently small ϵ, the content of α^b in Θ^b is $\langle \theta_1^b, \ldots, \theta_n^b \rangle$. We can then relate the eventual content of α^b to the congestion of α^b and thence to the congestion of α. Another example is the maze that Algorithm R computes for a typical trace θ. We do not explain this maze in terms of necessary crossings of diagonal cuts, but rather as the limit as $\epsilon \to 0$ of the maze for the ideal embedding of θ^b, deleting gates derived from trivial crossings. We can then argue that the ideal realization of θ is a tight track through this maze, and hence is computed by Algorithm R.

Rubber bands versus elastic chains

We relate the RBE of a sketch Σ to the plans of cuts and wires in the corresponding designs $\Theta^\flat$ by way of the elastic-chain equivalents of $\Theta^\flat$. An ECE of a design has a *structure* more abstract than its pure geometry: the segments of its chains and fringes intersect in a certain fashion, overlapping segments are sorted in a certain way, the segments of each chain are connected in a certain order and labeled with the wire they came from. One can show that these properties of the ECE settle, independent of which ECE one chooses for each value of ϵ. The structure of the RBE is a reflection of the settled structure of the ECE, and the latter can be recovered from the former.

The connection between rubber bands and elastic chains comes from the way we construct them. Recall how one constructs the rubber band for a trace. Let θ be a trace in the sketch $\Sigma = (\Xi, \Theta)$, and ıet Γ be a triangulation of the routing region of Σ by straight cuts. The images of the elements of Γ are called *doorways*. The trace θ passes through a certain sequence of doorways, which we "reduce" by eliminating consecutive occurrences of the same doorway. The resulting sequence $Im\,\gamma_1, \ldots, Im\,\gamma_n$, together with the terminals of θ, is a corridor for θ. The rubber band for θ the shortest path through this corridor. (We are assuming the correctness of Algorithm W.) We can construct the elastic chain for $\theta^\flat$ similarly. Let ϵ be small enough that $\Theta^\flat$ is a design on the sheet S_ϵ and $\Gamma^\flat$ is a set of disjoint cuts in S_ϵ. Because Γ is a triangulation, $\Gamma^\flat$ is a pattern of straight cuts, and the path code of $\theta^\flat$ in Γ settles to $\langle \gamma_1^\flat, \ldots, \gamma_n^\flat \rangle$. If necessary, displace a couple of initial and final segments of θ so that $\theta^\flat$ is free in $\Gamma^\flat$. (This change does not affect the rubber band of θ.) Then by Lemma 7c.1, the elastic chain for $\theta^\flat$ is eventually the shortest canonical path in S_ϵ from $\theta^\flat(0)$ to $\theta^\flat(1)$ whose seam list in $\Gamma^\flat$ is $\langle \gamma_1^\flat, \ldots, \gamma_n^\flat \rangle$.

Clearly the elastic chain and the rubber band are very similar. In the limit, the only difference is that the elastic chain is required to be canonical, whereas the parameterization of a rubber band is unimportant. One can show that for sufficiently small ϵ, the elastic chain approaches the same sequence of feature endpoints (within ϵ) that the rubber band touches, and that it passes left or right of them just as the rubber band does. In other words, the rubber band encodes the limiting structure of the elastic chain. This structure is all one needs in order to compute sequences of nontrivial crossings among cuts and elastic chains. (The trivial crossings may depend on which elastic chains one chooses, but the nontrivial crossings—the crossings in the link plans—do not.)

Use of the rubber-band equivalent

As previously mentioned, the primary task of the rubber-band equivalent is the computation of content. Given any straight cut α in the sketch $\Sigma = (\Xi, \Theta)$, it should compute a sequence $\langle \theta_1, \ldots, \theta_n \rangle$ of traces in Θ such that the content of $\alpha^\flat$ in $\Theta^\flat$ is eventually $\langle \theta_1^\flat, \ldots, \theta_n^\flat \rangle$. In the case of the condensed RBE, it should compute

$\sum_{i=1}^{n} \mathit{width}(\theta_i)$ instead. The former task subsumes the latter, so we concentrate on it alone. We take the case in which $\alpha^\flat$ is not a subpath of the elastic chain of any trace in Θ.

By Corollary 7c.6, it suffices for the RBE to compute the settled sequence of traces in the *cut plan* of $\alpha^\flat$ in $\Theta^\flat$, eliminating those corresponding to trivial crossings. Recall that this cut plan is just the sequence of crossings of $\alpha^\flat$ by elastic chains in $\Theta^\flat$, ordered by *precedence*. So we need to check two things: first, that it computes the correct number of nontrivial crossings of $\alpha^\flat$ by each elastic chain; and second, that it sorts the crossings according to precedence.

First we examine the problem of identifying the nontrivial crossings of $\alpha^\flat$. The following lemma gives us a handle on the problem.

Lemma 8c.1. *Let α be a straight cut in the sketch $\Sigma = (\Xi, \Theta)$ and let ρ be the elastic chain of a wire $\theta^\flat \in \Theta^\flat$. Eventually, a crossing (c, r) of $\alpha^\flat$ by ρ is trivial if and only if there are points $s \in I$ and $e \in \{0, 1\}$ such that $\rho_{r:s}$ is straight, $Im\, \rho_{s:e} \subset Bd\, S_\epsilon$, and $\rho(e)$ lies on a terminal of $\alpha^\flat$.* □

In the light of Lemma 8c.1, we consider the relationship between the crossings of a straight cut α, as computed by the RBE, and the trivial crossings of $\alpha^\flat$ by elastic chains in $\Theta^\flat$. Say $\alpha = p \triangleright q$, and let $r \triangleright s$ be a strand of the rubber band for some trace θ in Σ. If $\overline{pq}$ crosses $\overline{rs}$, there may eventually be a corresponding crossing between $\alpha^\flat$ and the *elastic chain* of $\theta^\flat$. Lemma 8c.1 tells us that such a crossing is eventually trivial if and only if either (a) $r \in \overline{pq}$ and $r \triangleright s$ is the first strand in its rubber band, or (b) $s \in \overline{pq}$ and $r \triangleright s$ is the last strand in its rubber band. With this in mind, we can see that the RBE correctly reports the (eventual) crossings of $\alpha^\flat$ in each case.

(1) If $\overline{rs}$ crosses the middle of $\overline{pq}$ and is not parallel to $\overline{pq}$, then the RBE reports a crossing.

(2) If the middle of $\overline{rs}$ intersects an endpoint of $\overline{pq}$, then the RBE reports a crossing provided that $\overline{rs}$ passes between p and q: it must leave p to the left if q lies on its right, and vice versa.

(3) If $\overline{rs}$ shares one endpoint with $\overline{pq}$, then the RBE reports a crossing provided that $\overline{rs}$ passes between p and q as before. In particular, no crossing is reported if the rubber band ends at the crossing point.

(4) If $\overline{rs} = \overline{pq}$, then the RBE reports a crossing provided that $\overline{rs}$ leaves p and q to opposite sides. In particular, $\overline{rs}$ must not be the first or last strand in its rubber band.

The other aspect of the RBE is they way it sorts the set of nontrivial crossings of the cut $\alpha^\flat$. When two crossings occur at different points along $\alpha^\flat$, which one precedes the other is obvious. When two crossings occur at the same point of $\alpha^\flat$, precedence is determined by which side of one chain the other chain approaches from. (See the end of Section 7C.) If one studies the procedure for constructing the RBE, one will see that the RBE defines precedence in the same way, but with rubber bands in place of elastic chains. Which elastic chain segments overlap, and which side of a chain another chain approaches from, depend only on the structures of those elastic chains. Since the rubber bands faithfully represent those structures, the RBE computes the right ordering within each cable for determining precedence.

Correctness of Algorithm T

Once we show that Algorithm T correctly computes the congestion of each critical cut, its overall correctness follows immediately from the sketch routability theorem (Theorem 8b.7). Let α be a critical cut in the sketch $\Sigma = (\Xi, \Theta)$. If α is a route for a trace $\theta \in \Sigma$, then α is the rubber band for θ. In this case Algorithm T detects no cables crossing the middle of α (since rubber bands do not cross over), and when querying the RBE for crossings at the endpoints of α, it also finds none (for the same reason). Thus Algorithm T deduces, correctly, that the congestion of α is zero. If α is not a route for any trace in Θ, then $\alpha^\flat$ is not a subpath of the elastic chain of any wire in $\Theta^\flat$. Then what Algorithm T gets is the limit as ϵ approaches 0 of sum of the widths of the wires in the content of $\alpha^\flat$. That sum *is just the flow across* $\alpha^\flat$ (see Lemma 7b.4). Since $\alpha^\flat$ is simple, Proposition 4b.6 says its flow is equal to its congestion, and its congestion stabilizes at that of α, by Proposition 8a.5. Therefore Algorithm T ends up with the congestion of α.

Mazes for ideal wires and traces

Next we move on to Algorithm R. Before explaining it, we must first show how to derive mazes for ideal wires. Recall from Section 7D that one obtains a maze in which an ideal wire ω is tight whenever for each diagonal slope $\pm\delta$ one has a pattern Γ with the following properties.

(1) Every cuts in Γ has angle $\pm\delta$.

(2) Every strut for ω is a subpath of some cut in Γ.

(3) For every seam $\gamma \in \Gamma$, the line containing γ separates the interiors of the pieces of Γ that include $Im\,\gamma$.

Such a pattern is easy to find, at least for sufficiently small ϵ, when ω is the ideal embedding of a wire $\theta^\flat$ derived from a trace in a sketch Σ. Given the angle δ, let Λ be a set of cuts that contains, for each diagonal cut α of Σ with $\dot{\alpha} = \pm\delta$, exactly one diagonal cut chosen from α and $\widehat{\alpha}$. Note that Λ cuts the routing region of Σ into triangles and trapezoidal strips. Assume ϵ is small enough that for all $\lambda \in \Lambda$

the path $\lambda^\flat$ cut in S_ϵ, and put $\Gamma = \Lambda^\flat$. Clearly $\Lambda^\flat$ satisfies conditions (1) and (3); an extension of Lemma 7d.1 shows that Γ also has property (2).

Now we consider the behavior of the ideal wire ω as ϵ approaches 0. Let θ be a trace in a sketch $\Sigma = (\Xi, \Theta)$ whose corresponding designs $\Theta^\flat$ are eventually $\sharp$-routable, and let ω be the ideal embedding of the wire $\theta^\flat$. Section 7D shows how to derive from the pattern $\Lambda^\flat$ and the elastic-chain equivalent of $\Theta^\flat$ a δ-tunnel for ω. It also shows, in Proposition 7d.6, that when these δ-tunnels are combined to form a maze, that ω is a tight track through that maze. If we compare this maze to the maze to the maze constructed by Algorithm R for θ, we find very few differences.

(1) The δ-tunnel for ω is derived from a path plan of the elastic chain for ω in $\Lambda^\flat$ and the cut plans of the cuts $\lambda^\flat$ in the elastic-chain equivalent of $\Theta^\flat$. The corridor for θ for the diagonal slope $\pm\delta$ is derived in the same way from the sequence of elements of Γ crossed over by the rubber band of θ and the sequences of rubber bands crossed over by the cuts in Γ.

(2) The δ-tunnel for ω accounts for the possibility that a crossing made by ω in $\Lambda^\flat$ may be trivial, and adjusts the gate endpoints accordingly. (See comments following equations (7–6) and (7–7).)

(3) A typical gate in the maze for ω is positioned with respect to the endpoints of a diagonal cut $\lambda^\flat$ using the norm $\|\cdot\|_\epsilon$, while the corresponding gate in the maze for θ (if one exists) is positioned with respect to the endpoints of λ *using the norm* $\|\cdot\|$.

All three differences essentially vanish in the limit. Only the first one is interesting; we dispose of the other two now. Difference (3) vanishes in the limit because $\|x\|_\epsilon \rightarrow \|x\|$ for all points x and $\lambda^\flat \rightarrow \lambda$ for all cuts λ as $\epsilon \rightarrow 0$. Difference (2) arises because the crossings found by the RBE, as discussed previously, are the nontrivial ones. The trivial crossings in the path plan for ω in $\Lambda^\flat$ are occur only at the beginning and end of the plan (Lemma 7b.3), and the gates corresponding to those crossings intersect the terminals of ω. (As ϵ approaches 0, those terminals shrink to points. Using these facts, one can check that the gates for ω derived from trivial crossings eventually cannot restrain ω, and thus can be removed from the maze for ω without ill effect.

Now we examine difference (1) more closely. A typical gate γ in the δ-tunnel for ω is constructed from a crossing of a cut $\lambda^\flat in \Lambda^\flat$ by the elastic chain ρ for ω. This crossing appears in two sequences: the wire plan of ρ in $\Lambda^\flat$, and the cut plan of $\lambda^\flat$ in the ECE of $\Theta^\flat$. (We need to choose an ECE that contains ρ.) Its position in the wire plan determines how many gates precede and follow γ in the δ-tunnel for ω, and its position in the cut plan determines which subpath of $\lambda^\flat$ the gate γ is to be; see equations (7–8) and (7–9) in Section 7D. Correspondingly, a typical doorway γ in the δ-corridor for θ is constructed from a crossing that appears in two plans: the sequence of that the rubber band of θ makes with cuts in Λ, and the sequence of crossings of rubber bands made by the diagonal cut λ. Its position in the first sequence determines its position in the corridor, and its position in the second sequence determines where the doorway is situated on λ; see equations (1-1) and (1-2) in Section 1D.

The two situations are in almost precise correspondence, if trivial crossings in the design model are ignored. In other words, I claim that the crossing sequences reported by the rubber-band equivalent, both of rubber bands and of cuts, faithfully represents the crossing sequences in the elastic-chain equivalent of the corresponding design, when trivial crossings are removed. This claim is an extension of previous statements about the role of the RBE. Together with the others, it implies that the maze for θ is the limit of the maze for ω.

There is another way to compute the maze for a trace, one that does not rely on the rubber-band equivalent. We noted in Section 7D that if gates from trivial crossings could be ignored, then one could easily compute the δ-tunnel for the ideal wire ω from an embedding Υ of the design $\Theta^\flat$ that conforms with $\Lambda^\flat$. This construction can be duplicated in the sketch model, as shown in Sections 9B and 9D. One constructs the *reduced intersection graph* of Λ and the sketch Σ, which represents all the crossings of cuts in Λ with wires in the embedding Υ, and connects them in the proper sequences. From these crossing sequences the limiting maze for a trace $\theta \in \Theta$ can be constructed quite easily. This idea was mentioned in Section 1E; it was discovered so recently that I have not had time to treat it in detail.

Correctness of Algorithm R

I now briefly outline the rest of the argument for the correctness of Algorithm R. The input to Algorithm R is a routable sketch $\Sigma = (\Xi, \Theta)$. By Lemma 8b.1, the corresponding designs $\Theta^\flat$ are eventually $\sharp$-routable, and hence for each trace $\theta \in \Theta$ the corresponding wire $\theta^\flat$ eventually has an ideal embedding ω. This ideal embedding is always a tight track in its maze. Any convergent sequence of these ideal embeddings, as ϵ approaches 0, must therefore converge to a path ρ through the limiting maze, which we have argued is the maze (or rather, the set of corridors) that Algorithm R constructs for θ. One can also show that ρ is a tight track through this maze, and is a realization of θ. Two consequences follow: that ρ is an ideal realization of θ, and that ρ can be reconstructed from the limiting maze as shown in Section 7E. The merging process described in Section 7E is exactly the one that Algorithm R performs. Hence Algorithm R constructs an ideal realization for each trace. Because the tight track through a maze is unique, the ideal realization is also unique. Thus the same argument also gives us a proof of Proposition 8b.5.

Chapter 9

Sketch Compaction

While the sketch routing problem is interesting in its own right, more interesting from a practical standpoint is its application to layout optimization. In this chapter I present a polynomial-time algorithm for the sketch compaction problem defined in Section 1A. Most of the text of this chapter appeared previously in my master's thesis [29], although some terminology has changed.

The importance of the sketch routing problem is limited by its assumptions, namely that the layer assignment and topology of the wires be predetermined. Most routing problems that arise in circuit design are multilevel problems with unspecified topology. One situation in which the layer assignment and topology are known, however, occurs when an existing layout is to be modified. A very common task in layout editing is that of making space in a layout for new components, or of compressing excess space from a layout. Mathematically the two tasks are very similar. I consider here the latter task, which is called **layout compaction**: given a layout, move its components to minimize the area it occupies. The sketch compaction problem is a specific formulation of the task of layout compaction, but it generalizes many of the compaction problems that are known to be solvable in polynomial time.

What follows is a brief discussion of compaction problems, the techniques commonly used to solve them, and the advantages of formulating the compaction problem in terms of sketches. A more formal introduction to the problem of sketch compaction is given in Section 9A.

Layout compaction

An automated compaction procedure is an effective tool for cutting the production costs of a VLSI circuit at low cost to the designer because the yield of fabricated chips is strongly dependent on the total circuit area. An effective compaction system also reduces design time by freeing the designer from continual concern over design rules. If excess layout space can be removed automatically, the designer can sketch a layout without making continual efforts to conserve area. For these reasons, compaction algorithms have gained widespread attention in the VLSI literature [16, 18, 24, 26, 58] and have been incorporated into many computer-aided circuit design systems, including [10, 16, 25, 47, 54, 57]. In many of these systems the input is symbolic, without explicit geometric content, and the function of the compactor is to convert the symbolic representation into a geometric specification. My compaction algorithm is not of this type. Its input and output are equally abstract: both are sketches.

Most compaction algorithms, including the one described here, compress a layout in one dimension only. To reduce both dimensions, the layout is alternately compacted horizontally and vertically until no further improvement can be found. Compaction in two dimensions simultaneously is NP-complete (although some 2-D compaction techniques may work well in practice [18]). Another common restriction is that the compaction algorithm cannot change the topology of any routing layer. In particular, its does not permit any component to jump over another component on the same layer. Without this restriction, the compaction problem again becomes NP-complete [26].

Constraint-based compaction

Many one-dimensional compaction systems [16, 25] use a **constraint-based** technique. To compact a layout horizontally, the program begins by assigning to each layout component i a variable x_i that represents the x-coordinate of the component's leftmost point. The **design rules** of the fabrication process are then used to derive constraints on the positions of the components. For example, if device i lies to the left of device j, and such devices must remain at least 2 units apart in order to function reliably, the compactor generates a constraint $x_j - x_i \geq 2 + w_i$, where w_i is the width of component i.

The design rules lead naturally to a set of constraints with nice properties. First of all, the constraints are not especially difficult to compute [24]. Second, they are sufficient to guarantee that the compacted layout is legal. Third, they are necessary if components cannot jump over one another. Fourth, the constraints are **simple linear inequalities**: they all can be represented in the form

$$x_j - x_i \geq a_{ij},$$

where x_i and x_j are two of the variables assigned to layout components, and a_{ij} is a constant.

Because of the simple form of the inequalities, they can be solved efficiently by graph-theoretic techniques. One constructs an edge-weighted graph in which the ith vertex represents the variable x_i, and in which an edge of weight a_{ij} from vertex i to vertex j represents the constraint $x_j - x_i \geq a_{ij}$. An assignment to the variables x_i that satisfies all the constraints is then determined by a longest-path computation on the graph. The resulting values specify the optimal positions of the components in the compacted layout. A good introduction to constraint-based compaction may be found in [18]; common algorithms for computing longest paths are discussed in [23]. (Most of the literature discusses the computation of shortest paths, but finding longest paths is equivalent to finding shortest paths when positive edge weights are replaced by negative, and vice versa.)

The computation of longest paths is especially efficient if the initial layout satisfies the design rules [30]. One writes all the constraints in terms of *displacements* of components from their original positions, rather than absolute coordinates. If d_i and d_j represent the horizontal displacements of modules i and j from their original positions, and $d_j - d_i \geq a_{ij}$ is a constraint, then the legality of the initial layout means that the inequality $d_j - d_i \geq a_{ij}$ holds when $d_j = d_i = 0$. In other words, the constant a_{ij} is nonpositive. Thus all the edges in the constraint graph have nonpositive weight, and so Dijkstra's algorithm may be used to compute the longest paths. (Usually Dijkstra's algorithm is used to find shortest paths, in which case the edge weights must be nonnegative, rather than nonpositive.) I use the same technique, writing constraints in terms of displacements, to speed up my compaction algorithm.

Automatic jog introduction

In order to perform any sort of compaction, the components of the layout must be differentiated into *modules*, which are fixed in size and shape, and *wires*, which are flexible. Common procedures for generating design-rule constraints [16, 18, 24] assume that wires are simply rectangular regions of variable height or width, and otherwise identical to modules. A vertical wire, for example, would be assigned an x-coordinate during horizontal compaction, and could only be moved rigidly from side to side. But one would often like a previously straight wire to bend around an obstacle during compaction, if the area of the circuit could thereby be reduced.

This problem is not easily overcome. Some systems [16, 57] attempt to solve it by allowing the designer to specify **jog points** at which wires may bend. In effect, the wires are broken into subwires at the jog points. Compaction then becomes an interactive procedure in which the designer repeatedly examines the compacted layout, adds more potential jog points, and retries the compaction operation. Other systems [16] attempt to insert jogs automatically, using ad hoc techniques which are not guaranteed to be effective. One technique that will work is to insert a jog point wherever a wire could possibly bend. If the wires are restricted to run in a grid, the number of such jog points can be made polynomial in the size of the input layout, since no wire need bend at a point far from a layout component. This technique, however, consumes large amounts of time and memory, and it does not generalize well to situations in which the grid is absent.

My approach to jog insertion involves replacing wires by routability constraints. Algorithm C, the sketch compaction algorithm, treats wires not as objects to be moved, but only as indicators of the topology of the layout. It constrains the positions of the modules to ensure that there exist routings of the wires, having the given topology, that form a proper sketch. It can express these **routability conditions** as simple linear inequalities and solve them as usual. When the optimal module placements have been established, the new wire paths are determined by a single-layer router such as Algorithm R.

This approach to compaction requires knowledge of necessary and sufficient conditions for routability. Ultimately these conditions are provided by the sketch routability theorem. The difficult part of Algorithm C is the construction of simple linear inequalities that capture the routability conditions. Fortunately, this construction is not especially model-dependent. In proving the correctness of Algorithm C, I have taken care to highlight the specific features of the sketch model on which it relies. Algorithm C is actually an implementation of a more abstract and general compaction technique that works in any model with the properties described in Section 9E.

Decomposition into planes

It remains to explain how sketch compaction, a single-layer problem, is germane to the compaction of layouts with multiple layers. Assuming that wires on different layers can be routed independently, then a multilayer compaction problem can be reduced to a set of single-layer problems, one per layer. One first computes the constraint systems for each layer independently. Since some modules extend into two or more layers, one must merge the resulting constraint systems by choosing the most restrictive constraint between every pair of modules. One then solves the merged system normally to place the modules, and routes the wires on each layer independently.

This procedure could generate illegal layouts if there were design rule constraints between wires on different layers. Fortunately, there are no problematic constraints in the most common VLSI technologies. In a standard nMOS process with one layer of metal, for example, the polysilicon and diffusion layers can be considered as one layer, or *plane* [47], for routing purposes, and metal the other plane. If transistors are considered to be modules, then the wiring in each plane contains no crossovers. Furthermore, wires on the two planes interact only at contact cuts, which are also represented as modules.

9A. Problem Statement

One definition of the sketch compaction problem was given in Section 1A. The input is a routable sketch with islands grouped into **modules**; each module is allowed to move horizontally as a unit. As modules move, traces must move as well in order to remain connected to their terminals. A sketch is *reachable* if it can be obtained from the input sketch by a continuous, piecewise linear motion that maintains routability. The sketch compaction problem is to find a proper, reachable sketch of minimum width. We will assume that two of the islands of the sketch are vertical lines, called **walls**, between which the other islands are to be squeezed. The width of the sketch is the euclidean distance between the two walls.

This definition is pleasantly simple and provides the right intuitions about sketch compaction, but we shall work from a second definition that is more tractable mathematically. The second definition is stated in terms of the *configuration space* of the sketch. One can prove that the two definitions are equivalent, though I will only prove one direction of the equivalence here.

Configuration space

The input sketch, call it S, is **modular**: its islands are grouped into modules, collections of features whose relative positions are fixed. The compactor is allowed to choose a horizontal displacement for each module. Such a vector of displacements is called a **configuration** of S. The configuration $\mathbf{d} = (d_1, \ldots, d_n)$ corresponds to a sketch $S(\mathbf{d})$ in which module i has been shifted right by a distance d_i (or left by a distance $-d_i$). Thus a configuration $\mathbf{d}$ determines how the features of $S(\mathbf{d})$ are to be placed; we shall consider the traces of $S(\mathbf{d})$ shortly. If the sketch S has n modules, then the set of all its configurations is the vector space R^n, and the origin $\mathbf{0}$ of this vector space corresponds to the original sketch. Convex combinations of configurations make sense: if $\mathbf{c}$ and $\mathbf{d}$ are configurations for S, then for each point $t \in [0,1]$ the vector $(1-t)\mathbf{c} + t\mathbf{d}$ is a configuration partway between $\mathbf{c}$ and $\mathbf{d}$.

Using configurations, we can describe how points on modules move during compaction. If p is a point in S, by which I mean that p lies on some feature of S, its x and y coordinates will be denoted x_p and y_p, respectively. The module in which p lies will be written $\mu(p)$, so the horizontal position of p in the configuration $\mathbf{d}$ is $x_p + d_{\mu(p)}$. The notation $p(\mathbf{d})$ stands for p shifted by $\mathbf{d}$, that is, the point $(x_p + d_{\mu(p)}, y_p)$. We also let $\Delta_{pq}(\mathbf{d})$ be difference in x-coordinates between $q(\mathbf{d})$ and $p(\mathbf{d})$, namely

$$\Delta_{pq}(\mathbf{d}) = (x_q + d_{\mu(q)}) - (x_p + d_{\mu(p)}).$$

To disallow the possibility of modules crossing over during compaction, we restrict attention to a subset of all configurations. Suppose p and q are points in S

having the same y-coordinate. If q lies to the right of p, then we only wish to consider configurations $\mathbf{d}$ in which $q(\mathbf{d})$ lies to the right of $p(\mathbf{d})$. So we let $\mathbf{C}(S) \subset R^n$ be the set of configurations $\mathbf{d}$ such that for all points p and q of S with $p_y = q_y$ and $p_x < q_x$, we have $\Delta_{pq}(\mathbf{d}) > 0$. We call $\mathbf{C}(S)$ the **configuration space** of the sketch S. The configuration space of S is convex, because it is the intersection of convex sets of the form

$$\{\, \mathbf{d} \in R^n : d_{\mu(q)} - d_{\mu(p)} > x_p - x_q \,\}, \qquad p, q \in S.$$

Only finitely many such constraints are needed to define configuration space: take the strongest one for each pair of modules. Hence $\mathbf{C}(S)$ is open in R^n.

Having defined the features of the sketch $S(\mathbf{d})$ as displaced copies of the features of S, we now define the traces of $S(\mathbf{d})$. The definition is somewhat arbitrary. We choose a continuous deformation of the plane, parameterized by the configuration $\mathbf{d}$, which moves each point horizontally and carries the features of S onto those of $S(\mathbf{d})$. We apply this deformation to the traces of S to obtain the traces of $S(\mathbf{d})$. More formally, if $h\colon R^2 \to R^2$ is a piecewise linear (PL) homeomorphism that is linear on each feature of a sketch S, let $h \diamond S$ denote the sketch obtained by replacing each feature P of S by $h(P)$ and replacing each trace θ of S by $h \circ \theta$. We choose any map $H\colon R^2 \times C(S) \to R^2$ with the properties of the following lemma, and put $S(\mathbf{d})$ equal to $H(\,\cdot\,, \mathbf{d}) \diamond S$. The islands and traces of $S(\mathbf{d})$ are assigned the same widths as the corresponding elements of S.

Lemma 9a.1. *Every modular sketch S admits a PL map $H\colon R^2 \times \mathbf{C}(S) \to R^2$ such that for every $\mathbf{d} \in \mathbf{C}(S)$ the map $H(\,\cdot\,, \mathbf{d})$ is a homeomorphism $h\colon R^2 \to R^2$ that carries each feature point p of S onto $p(\mathbf{d})$, factors as $h(x,y) = (h_y(x), y)$, and equals id_{R^2} if $\mathbf{d} = \mathbf{0}$.*

Outline of proof. Triangulate the routing region of S with cuts in such a way that each feature endpoint has two horizontal elements (features or cuts) incident upon it. For $\mathbf{d} \in \mathbf{C}(S)$, the map $h = H(\,\cdot\,, \mathbf{d})$ will take each triangle Δpqr in this triangulation to the triangle bounded by $p(\mathbf{d})$, $q(\mathbf{d})$, and $r(\mathbf{d})$, and will map the inside of the first triangle linearly onto the second. □

The choice of the map H does not affect the topology of $S(\mathbf{d})$. Suppose both F and G both have the properties given in Lemma 9a.1, let θ be a trace of S, and let $\mathbf{d} \in \mathbf{C}(S)$ be a configuration. Then the trace $G(\,\cdot\,, \mathbf{d}) \circ \theta$ is a route for $F(\,\cdot\,, \mathbf{d}) \circ \theta$: one homotopy between them is $(s,t) \mapsto \beta_t(s)$ where

$$\beta_t = F(\,\cdot\,, \mathbf{d}) \circ F(\,\cdot\,, t\mathbf{d})^{-1} \circ G(\,\cdot\,, t\mathbf{d}) \circ \theta.$$

Problem statement

Given a routable sketch S, one would ideally like to find a configuration $\mathbf{d} \in \mathbf{C}(S)$ such that $S(\mathbf{d})$ is routable, and can be routed in minimal width. But this problem is almost certainly NP-complete. The reason is that the routability conditions may not define a convex subset of configuration space, and hence the set of acceptable configurations $\{\, \mathbf{d} \in \mathrm{C}(S) : S(\mathbf{d}) \text{ is routable} \,\}$ can be very hard to search. For example, consider the sketch in Figure 9a-1. The set of acceptable configurations falls into two components: those in which the upper module lies entirely to the right of the lower module, and those in which the opposite is true. Intermediate configurations correspond to unroutable sketches, and thus the set of acceptable configurations is not convex. In most optimization problems, including compaction, one only expects to search a convex subset of the acceptable configurations in order to achieve a polynomial-time algorithm. Algorithm C searches the largest such region that contains the initial configuration, and thus finds the best configuration available to any algorithm of its type.

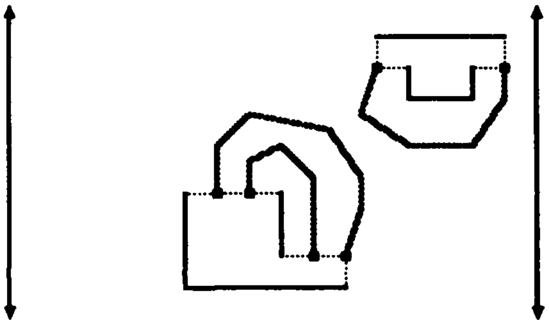


Figure 9a-1. *How traces can prevent modules from sliding past one another.* If the upper module is allowed to move to the left of the lower one, the set of acceptable configurations is not convex.

The new definition of the sketch compaction problem, and the one that we prove Algorithm C to satisfy, is the following. Given a routable sketch S, find a sketch $S(\mathbf{c})$ that has minimum width over all configurations $\mathbf{c}$ in the component of $\{\, \mathbf{d} \in \mathrm{C}(S) : S(\mathbf{d}) \text{ is routable} \,\}$ that contains $\mathbf{0}$, and output a proper realization of $S(\mathbf{c})$. By redefining sketch compaction, we have not made it easier; if anything, we have made it harder. Proposition 9a.2 implies that the output of Algorithm C is at least as good as the output of any algorithm that solves sketch compaction as originally defined.

Proposition 9a.2. *If T is any sketch reachable from a sketch S, then T is a realization of some sketch $S(\mathbf{d})$ where $\mathbf{d}$ lies in the component of $\{\, \mathbf{c} \in \mathrm{C}(S) : S(\mathbf{c}) \text{ is routable} \,\}$ that contains $\mathbf{0}$.*

Proof. Let T be reachable from S through a family $\{\, T(t) : t \in I \,\}$ of routable sketches, where $S = T(0)$ and $T = T(1)$. The definition of reachability requires that the deformation $t \mapsto T(t)$ be continuous and piecewise linear, that it move each module as a unit, and that no module move vertically. Several consequences follow. Each sketch $T(t)$ has its modules in the same position as $S(\mathbf{f}(t))$ for some configuration $\mathbf{f}(t) \in \mathrm{C}(S)$, and the function $\mathbf{f}: I \rightarrow \mathrm{C}(S)$ is continuous and piecewise linear. If for $t \in I$ the trace in $T(t)$ corresponding to a given trace θ in S is θ_t, then the map $(s,t) \mapsto \theta_t(s)$ is also piecewise linear.

We show that for $x \in [0,1]$, the sketch $S(\mathbf{f}(x))$ is a realization of $T(x)$, and therefore routable. The conclusion will then hold with $\mathbf{d} = \mathbf{f}(1)$: the path $\mathbf{f}$ runs from $\mathbf{0}$ to $\mathbf{d}$ in $\mathrm{C}(S)$, and its image is a connected subset of $\{\, \mathbf{c} \in \mathrm{C}(S) : S(\mathbf{c}) \text{ is routable} \,\}$.

Given $x \in [0,1]$, we find a continuous, piecewise linear deformation of $S(\mathbf{f}(x))$ into $T(x)$ that fixes their features. Let $H: R^2 \times \mathrm{C}(S) \rightarrow R^2$ be the map that defines the sketches $S(\mathbf{c})$ for $\mathbf{c} \in \mathrm{C}(S)$. The desired homotopy is

$$t \mapsto \left[H(\,\cdot\,,\mathbf{f}(x)) \circ H(\,\cdot\,,\mathbf{f}(tx))^{-1}\right] \diamond T(tx).$$

At $t = 0$ it equals $H(\,\cdot\,,\mathbf{f}(x)) \diamond T(0)$, which is $S(\mathbf{f}(x))$, and at $t = 1$ it becomes $T(x)$. Moreover, the homotopy is the composition of piecewise linear maps, and hence is piecewise linear. To say the same thing another way: Let θ be a trace in S, and for $t \in I$ let θ_t be the corresponding trace in $T(t)$. A bridge homotopy between θ_x and its counterpart $H(\,\cdot\,,\mathbf{f}(x)) \circ \theta$ in $S(\mathbf{f}(x))$ is $(s,t) \mapsto \beta_t(s)$ where

$$\beta_t = H(\,\cdot\,,\mathbf{f}(x)) \circ H(\,\cdot\,,\mathbf{f}(tx))^{-1} \circ \theta_{tx}. \qquad \square$$

9B. Computing Flows During Compaction

This section describes a procedure used to facilitate the computation of routability conditions for a sketch. Of course, the routability conditions are based upon the flows and capacities of cuts. (In this chapter I use flow as a synonym for congestion. If α is a cut, I write *flow*(α) in place of *cong*(α).) Capacities are purely geometric quantities, and can be computed from endpoint locations in constant time. In addition, they vary in a regular way with the movement of features during compaction. Flows, on the other hand, are topological quantities, and are relatively difficult to compute. Moreover, they depend in complex ways on the positions of features. Thus to compute flows, we require a data structure that captures the topology of the sketch and that is invariant under compaction. I begin by presenting such a structure. The proofs that justify this construction may be found in Section 9D.

Intersection graphs

The data structure we use is called the **adjacency graph** of the sketch. Its construction is straightforward, and is illustrated by Figure 9b-1. Given the input sketch S, first choose a finite set Γ of horizontal cuts, called **gates**, such that

(1) each island except the right wall contains the left endpoint of some gate, and

(2) each island except the left wall contains the right endpoint of some gate.

We call Γ a **partition** of S. The gates in Γ chop the routing region of S into simply connected pieces, each gate bordering on two pieces. (The routing region of S is the set of points between the two walls that lie on no feature of S.) With the addition of the gates in Γ, the sketch S forms a planar multigraph called the **intersection graph** of Γ and S. Its nodes are disjoint regions of the plane: the islands of S and the intervals of overlap between traces and gates where a trace *crosses over* a gate. Its arcs are the portions of gates and traces that connect the islands and intervals of overlap.

After constructing the graph of intersections between gates and traces, the next step is to reduce this graph, removing unnecessary crossings. In effect, one routes the traces of S so that they cross the gates in Γ as seldom as possible. Fortunately one need not construct the new traces; one need only remember crossings between traces and gates and the directions of those crossings. Wherever two nodes in the intersection graph are adjacent via both a gate and a trace, one removes whichever of those nodes represents a gate/trace crossing. (None, one, or two nodes will be removed.) To eliminate a node, one simply connects the incoming gate segments, replacing them by a single arc, and connects the incoming trace segments, replacing them by a single arc. After each removal the graph remains planar; one can imagine rerouting the affected trace to eliminate the unwanted crossings, without causing it to cross any other gates or traces. One repeats the operation of removing crossings until it can no longer be applied.

The graph that remains is called the **reduced** intersection graph of Γ and S. We mentioned it in Section 1E as a way of computing the content of the cuts in Γ. (In that application Γ contained all the diagonal cuts of a particular diagonal slope.) Each cut in Γ corresponds to a path in the reduced intersection graph, and the nodes internal to that path represent crossings with traces. If one replaces each such node by the corresponding trace, one obtains the content of the cut. I explain this fact further in Section 9D.

Adjacency graphs

The adjacency graph, which is actually a multigraph, is the planar dual of the reduced intersection graph. Since the direction of each crossing is known, one has enough information about the embedding of the intersection graph to construct its dual graph, call it G. A node of G corresponds to a face N of the intersection graph, a region of the plane. Let α be a simple path beginning at a feature point on the frontier of N. We say N **borders on $\boldsymbol{x}$ in the direction of $\boldsymbol{\alpha}$** if, among all the regions into which Γ partitions the routing region, N lies in the first one entered by α. (If

α lies entirely within a gate, we pretend that it enters the region directly above the gate.) If γ is a gate, we say that N **borders** γ if N has an arc representing adjacency across γ. We note for each edge of the G which gate or trace it crosses. We also build for each island P a data structure that lets us answer, in logarithmic time, queries of the form: Which nodes of G border on the point $p \in P$ in the direction of the straight cut $p \triangleright q$? There are at most two such nodes.

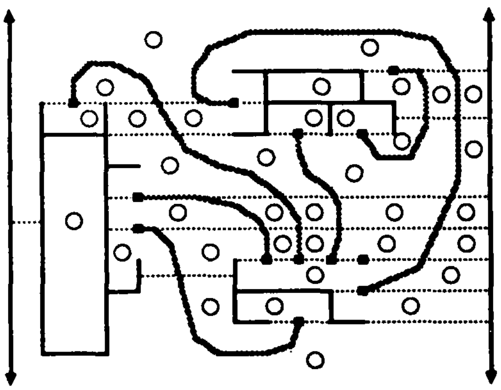


Figure 9b-1. *The adjacency graph of a sketch.* Dashed lines are gates, and circles are nodes of the adjacency graph. Wherever two such nodes are adjacent across a gate or trace, there is a *gate arc* or *trace arc*, respectively, in the adjacency graph. These arcs are omitted for clarity. Adjacency across features is not represented in the adjacency graph.

A key property of the adjacency graph is its invariance under horizontal compaction. During compaction, gates can only slide back and forth, and traces and features never cross over one another. Hence for any configuration $\mathbf{d} \in \mathrm{C}(S)$, the adjacency graphs of S and $S(\mathbf{d})$ are isomorphic. (The isomorphism preserves all relevant information up to change of configuration. For instance, it carries an arc representing adjacency across a gate $p \triangleright q$ to an arc representing adjacency across the gate $p(\mathbf{d}) \triangleright q(\mathbf{d})$. See Lemma 9d.1.)

Constructing the adjacency graph of a sketch is not difficult. Let $S = (F, W)$ be the input sketch. One must first choose a partition of S. One can do so very quickly by scanning; in any case, the cuts in the partition should be sorted by y-coordinate. One can then construct the graph of intersections between gates and traces by any natural method. Since every trace segment could cross every gate, this process might require $\Theta(|F|\,|W|)$ time and space, but will probably need much less. (Recall that $|D|$ denotes the size of the data structure D. Thus $|W|$ is not the number of traces, but rather the number of line segments that compose them.)

The remaining tasks—reducing the graph and taking its dual—also require at most $O(|F|\,|W|)$ time and space. I recommend the following event-driven method for reducing the intersection graph. First scan over the entire graph to identify the nodes to be removed, and place these in a queue. Then repeatedly take a node from the queue, delete it, check whether any of its neighbors have become removable due to the newly created arcs, and if so, add them to the queue. The checking can be performed in constant time because nodes that represents crossings have only four incident arcs. Hence each arc of the intersection graph is examined only a constant number of times before the queue becomes empty. Building the dual graph is straightforward; one simply walks around the faces of the original graph, creating dual nodes and arcs as necessary. The time and space taken by this construction are both proportional to the size of the dual graph.

Searching the adjacency graph

The purpose of the gates is to relate cuts in the sketch to the sketch topology. Though we work nearly always with straight cuts, I explain the application of the adjacency graph to a somewhat wider class of cuts. If a cut makes no crossings with gates that are removable by a bridge homotopy, then we can use the sequence of gates crossed by the cut, its **gate list**, to search through the adjacency graph and compute its flow. Let Γ be a partition of S, and let β be a bridge in S. We say β is **direct** if

- it never intersects the middle of a gate without crossing over,
- it intersects each gate at most once, and
- no trace terminal intersects both β and a gate crossed by β.

Every straight cut that is not a gate is direct.

Paths in the adjacency graph, too, correspond to sequences of gate crossings. To see how, notice that there are two kinds of arcs in the adjacency graph: **trace arcs**, which represent adjacency across a trace, and **gate arcs**, which represent adjacency across a gate. A path ξ through the adjacency graph thus crosses a sequence of traces, one for each trace arc in ξ, and a sequence of gates, one for each gate arc in ξ. The sequence of gates that ξ crosses is the **gate list** of ξ. We assign each trace arc a **length** equal to the width of the corresponding trace; gate arcs have length 0. Then every path in the adjacency graph has a nonnegative length. It turns out that the flow across a cut with gate list ξ is directly related to the lengths of paths in the adjacency graph with gate list ξ.

Proposition 9d.4. The flow across a direct cut α in the sketch S is the length of the shortest path in the adjacency graph of S that

(1) begins at a node bordering on $\alpha(0)$ in the direction of α,
(2) ends at a node bordering on $\alpha(1)$ in the direction of $\widehat{\alpha}$, and
(3) has gate list equal to that of α. □

It also turns out that the shortest path with a given gate list may be found by a greedy algorithm: first take the shortest path to the first gate, cross it, find the shortest path from there to the second gate, and so on. Algorithm F computes the flow across a straight cut using this greedy method. It repeatedly searches through the **skeleton** of the adjacency graph G, the subgraph consisting of the nodes and trace arcs of G. Because the gates partition the routing region into simply connected pieces, the skeleton of G is a forest T. The input to Algorithm F is the gate list of a direct cut. Algorithm F also works if the input cut is a gate, provided that its gate list is considered to be empty. See Proposition 9d.5 for a correctness proof.

Algorithm F. (Computes the flow across a straight cut.)

```
Input: a direct cut $\alpha$ with gate list $\langle \gamma_1, \ldots, \gamma_n \rangle$; the adjacency graph $G$ with
        skeleton $T$.
Output: the flow across $\alpha$.
Local variables: integers $i$ and $t$; nodes $u$, $v$, $x$, and $y$.
 1. **return** min{ DIST$(x,y)$ : $x$ borders on $\alpha(0)$ in the direction of $\alpha$,
          and $y$ borders on $\alpha(1)$ in the direction of $\widehat{\alpha}$ }.
 2.    **function** DIST$(x,y)$;
 3.    $t \leftarrow 0$; $u \leftarrow x$;
 4.    **for** $i \leftarrow 1$ **to** $n$ **do**
 5.       $v \leftarrow$ a node bordering $\gamma_i$ that is closest to $u$ in $T$;
 6.       $t \leftarrow t +$ the distance from $u$ to $v$ in $T$;
 7.       $u \leftarrow$ the node adjacent to $v$ across the gate $\gamma_i$;
 8.    **return** $t +$ the distance from $u$ to $y$ in $T$.
```

Data structures for Algorithm F

The most time-consuming steps of Algorithm F involve searching through the skeleton T of the adjacency graph. If simplicity is desired, one may implement lines 5, 6, and 8 of Algorithm F using Dijkstra's shortest path algorithm. This approach may work well in practice, but its worst-case behavior is poor; it could require $\Omega(n\,|T|)$ time on a gate sequence of length n. This section shows how the adjacency graph G may be preprocessed so that Algorithm F takes $O(\log^2 |G|)$ time per iteration. The preprocessing requires $O(|T| \log^2 |G|)$ time and creates data structures that occupy $O(|T| \log |T|)$ space. One could speed up Algorithm F even further by precomputing the distance between every pair of nodes in T, but only at the cost of $\Omega(|T|^2)$ space.

We speed up the searches by taking advantage of the fact that T is a forest. For convenience we may assume that T is a tree, since each search uses only one component of T. The first task is to preprocess T so that one can quickly determine the distance between any pair of its nodes, thereby speeding up lines 6 and 9 in Algorithm F. The second task is to preprocess T so that one can compute efficiently the closest node in a connected subset of T to a given node. This ability is sufficient to implement line 5 of Algorithm F, because in each component of T, the set of nodes bordering a gate is connected.

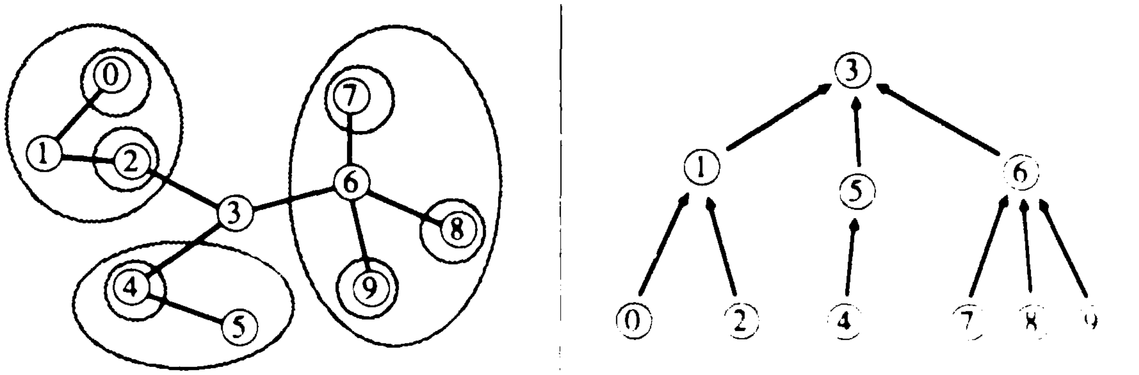


Figure 9b-2. *A tree, drawn with solid lines, and its decomposition tree*

The idea is to decompose the graph T recursively, forming a **decomposition tree** D on the nodes of T. Figure 9b-2 shows the construction. Let n be the number of nodes in the graph T. The separator theorem for trees [27] implies that T contains a node r whose removal disconnects T into subtrees containing at most $\frac{2}{3}n$ nodes each. Moreover, we can find the node r in linear time using depth-first search to compute the sizes of subtrees. Now we decompose the subtrees of r recursively, obtaining a decomposition tree for each one. The roots of these trees become the children of r in the decomposition tree D for T. At each stage of the decomposition, the sizes of subtrees are reduced by a constant factor, so D has height $\Theta(\log |T|)$. The recursive construction of D takes $\Theta(|T| \log |T|)$ time because it examines each node and arc in T just $O(\log |T|)$ times. We store at each node of T its distance in T from each of its ancestors in D. These distances can be computed in $\Theta(|T| \log |T|)$ time during the construction of D, and their storage requires $\Theta(|T| \log |T|)$ space.

This information allows one to compute quickly the distance between two nodes of T. One simply finds their lowest common ancestor (LCA) in D and then adds their distances (in T) from that ancestor. This procedure takes at most $O(\log |T|)$ time; it works because the LCA in D of two nodes in T either equals one of them or separates them in T.

Some extra preprocessing is needed before we can compute lowest members of connected sets of T. Let D be the same decomposition tree as above. We can compute in advance the LCA's of the connected sets that we care about. There are $O(|F|)$ such sets, one for each gate: the set of nodes in T bordering on that gate. The LCA in D of a connected set $C \subseteq T$ is a member of C and can be computed in $O(\log |G| \log |T|)$ steps if some node of C is known, assuming that membership in C can be tested in $O(\log |G|)$ time.* Thus the preprocessing of the connected

* In practice, membership in C could probably be tested in $O(1)$ time, since each

sets uses $O(|F|\log|G|\log|T|)$ time and $O(|F|)$ space. We also store, for each node y and for each of its ancestors x, the highest node in D that is interior to the to the path in T between x and y. In case x and y are adjacent in T, we store *nil* instead. To produce this information requires $O(|T|\log|T|)$ space and $O(|T|\log^2|T|)$ time.

Algorithm V: Finds the closest node in a connected set $C \subseteq T$ to a node u.
Additional input: uses the data from preprocessing in lines 2 and 4.
Local variables: nodes v and z.

```
1. if $u \in C$ then return $u$
2. $v \leftarrow$ LCA$(C)$
3. if $v$ is not an ancestor of $u$ then $u \leftarrow$ LCA$(u, v)$
   repeat
4.     $z \leftarrow$ the highest node in $D$ on the path in $T$ between $u$ and $v$
5.     if $z = nil$ then return $v$
6.     if $z \in C$ then $v \leftarrow z$ else $u \leftarrow z$
   until false
```

Unlike Algorithm F, the correctness of Algorithm V does not depend on any topological facts, so we prove it here. First we show that before every iteration of the loop the following invariants hold: (a) $v \in C$ but $u \notin C$; (b) one of u and v is an ancestor of the other; and (c) the closest node in C to u is the closest node in C to the original input u. Lines 1–3 serve to establish these invariants. Line 3 does not harm invariant (c), for if v is not an ancestor of u, then the LCA of u and v is on every path between u and C. Hence the closest node in C to u is also the closest node in C to LCA(u, v). Now we check that the loop maintains the invariants. Invariant (a) is maintained by line 6. That line does not affect invariant (b) either, because u, v, and z are all on the same branch of D. Invariant (c) can only be affected if $z \notin C$. But in that case, every path from u to C passes through z because C is connected. Hence the closest node in C to u is also the closest node in C to z. Finally, line 4 outputs the correct node: z being *nil* means that u and v are adjacent, which makes v the closest node in C to u. Therefore when Algorithm V terminates it produces the correct answer.

It remains to bound the number of iterations of the loop. We bound it by the height of D, by showing that the height of the node z in D decreases by at least one at each step. Let u_0, v_0, and z_0 be the values of u, v, and z at one iteration, and let

node of T would probably have a bit vector to specify which gates it borders. From a theoretical point of view, this approach is no good because it requires $O(|F||G|)$ bits of storage. If the arc list of each node of G is kept in an appropriate data structure, then it takes at most $O(\log|G|)$ time to determine whether a node is adjacent to a particular gate. This method loses no asymptotic space efficiency.

u_1, v_1, and z_1 be their values at the next iteration. The path between u_1 and v_1 is a subpath of the path between u_0 and v_0, so z_1 is no higher than z_0. Suppose they were the same height. Then $LCA(z_0, z_1)$ would be a higher node separating z_1 from z_0. This node would be on the path between u_0 and v_0, contradicting the definition of z_0. Therefore z_1 is strictly lower than z_0. We conclude that Algorithm V runs in $O(\log |T|)$ iterations. Each iteration requires $O(\log |G|)$ time due to the membership test in line 6. Hence Algorithm V finishes in $O(\log |G| \log |T|)$ time.

9C. The Compaction Algorithm

Having at our disposal a subroutine for computing flow, we now develop an algorithm for solving the sketch compaction problem.

Potential cuts

The basic notion underlying the compaction algorithm is that of a **potential cut**. Let P and Q be features in the original sketch S, and let ψ be a continuous, piecewise linear function that defines, for each configuration $\mathbf{d}$ in $\mathbf{C}(S)$, a line segment between the features $P(\mathbf{d})$ and $Q(\mathbf{d})$ in the sketch $S(\mathbf{d})$. The function ψ is a potential cut if the position of $\psi(\mathbf{d})$ relative to $P(\mathbf{d})$ and $Q(\mathbf{d})$ depends only on the displacement between $P(\mathbf{d})$ and $Q(\mathbf{d})$, namely $\Delta_{PQ}(\mathbf{d}) = d_{\mu(Q)} - d_{\mu(P)}$ (stretching the notation slightly). In other words, ψ must satisfy the following condition.

Potential cut property. If two configurations $\mathbf{d}$ and $\mathbf{d}'$ satisfy $\Delta_{PQ}(\mathbf{d}) = \Delta_{PQ}(\mathbf{d}')$, then $\psi(\mathbf{d}')$ is equal to $\psi(\mathbf{d}')$ shifted to the right by $d'_{\mu(P)} - d_{\mu(P)}$ units.

The **capacity** of a potential cut ψ in the configuration $\mathbf{d}$, denoted $cap(\psi(\mathbf{d}))$, is defined as $\|\psi(\mathbf{d})\|$ minus the average of the widths of the islands that contain the endpoints of $\psi(\mathbf{d})$. This definition agrees with the usual one when $\psi(\mathbf{d})$ is a cut. The configuration $\mathbf{c}$ **protects** a potential cut ψ unless $\psi(\mathbf{c})$ is an unsafe, nonempty cut. The significance of these definitions lies in a reformulation of the sketch routability theorem in terms of potential cuts.

Definition 9c.1. Let S be a sketch, and let $\mathbf{c} \in \mathbf{C}(S)$ be a vector in its configuration space. For every endpoint p of a feature in F, and for every other feature Q in S, let $\chi_{pQ}(\mathbf{c})$ denote the linear path from $p(\mathbf{c})$ to the closest point on $Q(\mathbf{c})$, measured in the wiring norm with tiebreaking in the euclidean norm. Then χ_{pQ} is a potential cut for S, which we call **critical**.

Theorem 9c.2. *Let $\mathbf{c}$ be a configuration of a routable modular sketch S. The sketch $S(\mathbf{c})$ is routable if and only if $\mathbf{c}$ protects every critical potential cut of S.*

Theorem 9c.2 follows directly from Theorem 8b.7.

Algorithm overview

Algorithm C works by finding a subset of configuration space, determined by simple linear inequalities, whose configurations protect every critical potential cut. The subspace searched is chosen so as to include the initial configuration.

The central problem is to find a simple linear inequality that ensures that a potential cut, say ψ, is protected. We may assume that ψ connects features in different modules, for otherwise ψ cannot generate a useful constraint. So $\psi(\mathbf{d})$ is not empty in any configuration $\mathbf{d}$, and it is enough to ensure that $\psi(\mathbf{d})$ is safe whenever it is a cut. One would like to use the routability condition $cap(\psi(\mathbf{d})) \geq flow(\psi(\mathbf{d}))$ as a constraint on the configuration $\mathbf{d}$, but for most potential cuts ψ, this constraint is not a simple linear inequality. The difficulty lies not with the capacity of $\psi(\mathbf{d})$, which is determined solely by the geometry of $S(\mathbf{d})$, and depends in a simple way on the displacements d_i. Rather, the quantity $flow(\psi(\mathbf{d}))$ is hard to characterize, because it depends on the relation of the line segment $\psi(\mathbf{d})$ to the topology of the sketch $S(\mathbf{d})$.

The solution is to find a specific configuration $\mathbf{c}$ such that whenever the potential cut $\psi(\mathbf{d})$ is unsafe, its flow is equal to $flow(\psi(\mathbf{c}))$. The constraint $cap(\psi(\mathbf{d})) \geq flow(\psi(\mathbf{c}))$ is then sufficient to protect ψ. Moreover, when this constraint is written in terms of the variables d_i, it becomes a disjunction of two simple linear inequalities, because the right hand side is constant. Because we care only about configurations reachable from $\mathbf{0}$, one of the two inequalities can be discarded. To find $\mathbf{c}$, the algorithm looks for a configuration that minimizes the capacity of ψ, subject to the condition that all critical cuts of smaller vertical span are protected. These shorter cuts force the other features to the side of ψ on which they must lie if ψ is ever to become unsafe. If, in this way, the algorithm finds a configuration $\mathbf{c}$ that does not protect ψ, then the routability condition for $\psi(\mathbf{c})$ is remembered. Otherwise, the potential cut ψ is ignored.

Description of the compaction algorithm

Since critical cuts move in nontrivial ways during compaction, Algorithm C considers two more types of potential cuts as well. For each pair (p, q) of feature endpoints, we consider the potential cut ϕ_{pq} defined by $\phi_{pq}(\mathbf{d}) = p(\mathbf{d}) \triangleright q(\mathbf{d})$. We also consider a potential cut ϕ_{pq} for every horizontal cut $p \triangleright q$ incident on a feature endpoint. We throw away a potential cut ϕ_{pq} or χ_{pQ}, however, if p lies in the same island as q or Q, because such cuts cannot generate any useful constraints.

Algorithm C processes the potential cuts in a particular order. First come the horizontal potential cuts; these generate the constraints that prevent features from crossing over one another. Next come the potential cuts ϕ_{pq} between feature endpoints, in order of increasing **height**. The height of ϕ_{pq} is the quantity $|y_q - y_p|$.

Finally the algorithm considers the critical cuts χ_{pQ}, also in order of increasing height. The height of χ_{pQ} is determined as follows. For $\mathbf{d} \in \mathrm{C}(S)$, whether the endpoint $\chi_{pQ}(\mathbf{d})(1)$ has y-coordinate greater than, equal to, or less than that of $\chi_{pQ}(\mathbf{d})(0)$ is independent of $\mathbf{d}$. We call χ_{pQ} "upward", "downward", or "horizontal" accordingly. The height of χ_{pQ} is defined as $|y_q - y_p|$, where q is

(1) the point on Q with greatest y-coordinate, if χ_{pQ} is upward;

(2) the point on Q with least y-coordinate, if χ_{pQ} is downward;

(3) the point $\chi_{pQ}(\mathbf{d})(1)$, for any $\mathbf{d}$, if χ_{pQ} is horizontal.

This definition ensures that whenever a path of the form $\chi_{rQ}(\mathbf{d})$ is a subpath of $\chi_{pQ}(\mathbf{d})$, that either $\chi_{rQ}(\mathbf{d}) = \phi_{rq}(\mathbf{d})$ for some feature endpoints r and q, or else the height of χ_{pQ} exceeds that of χ_{rQ}.

Algorithm C maintains a system I of simple linear inequalities among the displacements d_i, represented as an edge-weighted graph over the modules. The result of processing a potential cut ψ is a simple linear inequality which, when added to I, ensures that all configurations satisfying the constraints in I protect ψ. Initially I is empty. After Algorithm C processes all the potential cuts, the constraint system I is complete, and the algorithm solves it using a longest-path algorithm. The resulting configuration is used to build an output sketch, which is then routed using Algorithm R (thus minimizing trace lengths).

To process a potential cut ψ between the features P and Q, Algorithm C examines how ψ varies with the relative positions of P and Q. By the definition of potential cut, the length $\|\psi(\mathbf{d})\|$ of ψ is some function l of $d_{\mu(Q)} - d_{\mu(P)}$. And for all potential cuts we use, this function l is *convex* (see Lemma 9f.2). Let Δ denote a point at which l takes on its minimum value. By the symmetry between P and Q, we may assume $0 \geq \Delta$. Algorithm C computes the constraint for ψ thus. First it solves the current constraint system I together with the additional constraint $d_{\mu(Q)} - d_{\mu(P)} \geq \Delta$, fixing $d_{\mu(P)}$ and minimizing $d_{\mu(Q)}$. Call the resulting configuration $\mathbf{c}$. If $\mathbf{c}$ protects ψ, then the algorithm does nothing further with ψ. Otherwise it computes the largest value Δ^+ such that $l(\Delta^+) = \mathit{flow}(\psi(\mathbf{c}))$, and adds to I the constraint

$$d_{\mu(Q)} - d_{\mu(P)} \geq \Delta^+. \tag{9–1}$$

We call inequality (9–1) the *constraint derived from ψ in configuration* $\mathbf{c}$. The complexity of computing the quantities Δ and Δ^+ depends only upon the wiring norm, and so I treat it as constant.

If ψ is horizontal, constraint solving is unnecessary because the flow across ψ is independent of configuration. In this case Algorithm C simply adds to I the constraint derived from ψ in the configuration $\mathbf{0}$.

The compaction algorithm is summarized below. It assumes that the left and

Algorithm C. (Compacts a sketch horizontally.)
Input: a sketch $S = (F, W)$ with n modules specified.
Output: a proper, compacted sketch.
Local variables: the points p and q, a configuration $\mathbf{c}$, the constraint graph I over variables d_i $(1 \leq i \leq n)$, coordinate value Δ.
Subroutines: Algorithm F is used to compute flows in lines 2 and 9; Dijkstra's algorithm is used in lines 7 and 10; Algorithm R is used in line 11

```
 1. Preprocess $S$ as described in Section 9B;
 2. Let $I$ be the set of constraints derived from the horizontal cuts in the initial
        configuration $\mathbf{0}$;
 3. foreach other potential cut $\psi$ between features $P$ and $Q$, in order, do
 4.   if $\mu(P) \neq \mu(Q)$ then
 5.     Find $\Delta$ such that $\|\psi(\mathbf{d})\|$ is minimal when $\Delta_{PQ}(\mathbf{d}) = \Delta$.
 6.     if $\Delta > 0$ then exchange $P$ and $Q$ and negate $\Delta$.
 7.     Find a configuration $\mathbf{c}$ that minimizes $c_{\mu(Q)} - c_{\mu(P)}$ while obeying the
            constraints $I \cup \{\Delta_{PQ}(\mathbf{d}) \geq \Delta\}$.
 8.     if $\psi(\mathbf{c})$ is a cut in $S(\mathbf{c})$ then
 9.       if $flow(\psi(\mathbf{c})) > cap(\psi(\mathbf{c}))$
            then add to $I$ the constraint derived from $\psi$ in the configuration $\mathbf{c}$.
10. Find a configuration $\mathbf{c}$ satisfying $I$ that minimizes $c_n - c_1$.
11. Route the sketch $S(\mathbf{c})$ and output the result.
```

right walls of the sketch compose modules 1 and n, respectively

The idea behind the algorithm

Why does Algorithm C work? The correctness of Algorithm C rests on two facts about the configuration $\mathbf{c}$ found at line 7. If $\mathbf{c}$ protects the potential cut ψ, then so do all configurations that satisfy the constraints already generated. And if $\mathbf{c}$ does not protect ψ, then any other configuration $\mathbf{d}$ that satisfies the previously generated constraints but fails to protect ψ assigns ψ the same flow as $\mathbf{c}$ does. We give these statements formal proof in Section 9F, but the argument can be outlined here.

Consider moving linearly from $\mathbf{c}$ to another configuration $\mathbf{d}$. By convexity of the constraints, all the intermediate configurations $\mathbf{b}$ satisfy the existing constraints. Among these, consider the configurations in which ψ is actually a cut. These form a set of open intervals in the line segment between $\mathbf{c}$ and $\mathbf{d}$. Within each interval the flow across $\psi(\mathbf{b})$ is constant, because the trace code of $\psi(\mathbf{b})$ is unchanged until $\psi(\mathbf{b})$ crosses some feature. And if $\mathbf{b}$ is a point where $\psi(\mathbf{b})$ ceases to be a cut, an endpoint of an interval, then the flow prevailing in that interval is at most the capacity of $\psi(\mathbf{b})$. (Here we rely on the results of Section 4F concerning chains of cuts.) Hence $\psi(\mathbf{d})$ can be unsafe only if the capacity of ψ attains a local minimum in the interval containing $\mathbf{d}$. Because the capacity of $\psi(\mathbf{b})$ is a convex function of $\mathbf{b}$, there is at most one such interval. And if there is one, it contains $\mathbf{c}$, because $\mathbf{c}$ was chosen to minimize $cap(\psi(\mathbf{c}))$.

Details of the implementation

The computation of flows in line *2* is performed using Algorithm F of the previous section. The horizontal cuts themselves may be found by any convenient method, as the algorithm's run time will be dominated by other factors.

Line 3 requires that the potential cuts ϕ_{pq}, where p and q are feature endpoints, be enumerated in increasing order of height. Writing down and sorting all pairs of feature endpoints would waste large amounts of space; the following approach is better. First sort the feature endpoints by y coordinate, and associate with each endpoint the next higher endpoint. Place these pairs in a priority queue, and keep the queue sorted by difference in y coordinates. At each iteration of the loop (lines 3-9), withdraw the best element $\{p, q\}$ from the priority queue and process the potential cut ϕ_{pq}. Then find the next endpoint q' above q in y coordinate, if one exists, and insert the pair $\{p, q'\}$ into the priority queue. This method uses linear space, and no more time than other parts of Algorithm C. A similar method may be used to enumerate the critical potential cuts ψ_{pq} in order of height.

To solve the constraint system in line 7, it suffices to compute longest paths from the vertex $\mu(I')$. Dijkstra's algorithm can be used for the purpose, because every edge in the graph has weight *zero* or less. (Normally, Dijkstra's algorithm is used to find shortest paths, and then the edge weights must be nonnegative.) To see why edge weights are nonpositive, consider the case when all the displacements d_i are zero. Using the assumption that the initial configuration is legal, one can prove that it obeys all constraints. Hence if $d_j - d_i \geq a_{ij}$ is a constraint in I, then it holds under the assignment $\mathbf{d} = \mathbf{0}$. The result is that $0 - 0 \geq a_{ij}$, that is, a_{ij} is nonpositive.

Once the algorithm finds the key configuration $\mathbf{c}$ in line 7, line 8 must determine whether $\phi_{pq}(\mathbf{c})$ is a cut. To do so it tests all features in $S(\mathbf{c})$ for intersection with $\phi_{pq}(\mathbf{c})$.

Line 9 invokes Algorithm F to calculate $flow(\phi_{pq}(\mathbf{c}))$. It requires as input the trace code of $\phi_{pq}(\mathbf{c})$, which can be found by checking every gate that lies between y_p and y_q in y-coordinate. Include only those gates of $S(\mathbf{c})$ that cross $p(\mathbf{c}) \triangleright q(\mathbf{c})$. The gate sequence should, of course, be sorted by y-coordinate, and all crossings must be from bottom to top. Presorting all the gates by y-coordinate eliminates the need to sort each individual trace code.

In line 10, Dijkstra's algorithm should be used once again, this time computing longest paths in I from module 1, which is the left edge of the bounding box of the sketch. If desired, the designer or design system may add other simple linear inequalities to I provided that they are all satisfied by the initial layout $S(\mathbf{0})$.

The configuration $\mathbf{c}$ found in line 10 specifies the optimal compacted sketch, but that sketch must still be constructed at line 11. For the purpose of applying Algorithm R, it is not necessary to construct a complete sketch $S(\mathbf{c})$, but only to produce the rubber band equivalent (RBE) of $S(\mathbf{c})$. The features of the RBE are the same as those of $S(\mathbf{c})$ and can be located easily. The rubber bands can be found as follows. The set of points not lying on features or gates of $S(\mathbf{c})$ is a simply connected region, and its boundary is polygonal (if we allow vertices at infinity). Hence it can be triangulated quickly, and the resulting set of triangles forms a tree under the obvious adjacency relation. We can therefore find for each trace θ in $S(\mathbf{c})$ the shortest sequence of triangles that a realization of θ could pass through, and apply Algorithm W from Section 1B to find the rubber band of θ.

Complexity analysis

The worst-case time complexity of Algorithm C is $O(|S|^4)$. (Recall that $|S|$ is the size of the data structure S. If $S = (F, W)$, then $|S| = |F| + |W|$.) We can obtain a more precise bound, however, in terms of $|G|$ and $|I|$. What follows is a line-by-line breakdown of time costs.

(1) According to Section 9B, the preprocessing phase takes time $O(|F||W| + |G|\log^2|G|)$, where G is the adjacency graph of the input sketch (F, W).

(2) Computing constraints for horizontal cuts is no harder than computing them for the other cuts, so line 2 may be ignored.

(3) Enumerating pairs of feature endpoints takes $O(|F|^4 \log|F|)$ time, $O(\log|F|)$ time per pair. This quantity is dominated by other terms.

(7) The primary bottleneck is the call to Dijkstra's algorithm in line 7; it runs in time $O(|E| + |V|\log|V|)$ on a graph (V, E) [12]. Since $|E|$ is $O(|I|)$, and $|V|$ is n, the number of modules, line 7 uses $\Theta(|I| + n\log n)$ time in each of $O(|F|^4)$ iterations.

(8) Line 8 takes $O(|F|)$ time per potential cut, and hence line 9 dominates it.

(9) Algorithm F uses $O(|F|\log^2|G|)$ time, so line 9 costs $O(|F|^3\log^2|G|)$ time in total.

(11) Careful analysis shows that the construction and routing of the output sketch requires only $O(|F||G|\log|G|)$ time.

The contributions of the remaining lines are negligible. Thus the total running time of Algorithm C is

$$O(|F||W| + |G|\log^2|G| + |F|^4(|I| + n\log n) + |F|^3\log^2|G| + |F||G|\log|G|).$$

Since $|I| = O(|F|^2)$ and $|G| = O(|F|\,|W|)$, this expression yields the claimed bound of $O(|S|^4)$. The only term that exceeds $O(|S|^3 \log^2 |S|)$ is the term $|F|^2\,|I|$ due to repeated constraint solving at line 7.

Which part of Algorithm C will dominate in practice is not clear. In the worst case, $|G|$ can be as high as $\Omega(|F|\,|W|)$, if some $\Omega(|W|)$ trace segments make necessary crossings with $\Omega(|F|)$ gates each. In most situations, however, $|G|$ should be closer to $|F|$. Making reasonable estimates about the average run time of Algorithm F and the density of the constraint graph I, one can predict that actual performance for the entire operation will probably approach $\Theta(|F|^{3+\epsilon})$ for some small positive value of ϵ.

Space usage is easier to evaluate: the main contributors are the graphs G and I, along with Algorithm R, which may use $O(|F|\,|G|)$ space in the worst case. Thus the worst case bound is $O(|F|^2\,|W|)$, but none of the data structures of Algorithm C or Algorithm R is likely to approach its maximum size. The actual figure will depend on the number of crossings between traces and certain cuts in the sketch (e.g., gates), and will probably look like $\Theta(|F|^{1+\alpha})$ for some constant $\alpha \in (0, 1)$.

9D. The Adjacency Graph of a Sketch

We begin our study of Algorithm C at its foundations: the definition of the adjacency graph of a sketch, and how the sketch itself varies with configuration. We prove that both are well defined, and relate the adjacency graph to the flows across cuts. The goal of this section is to prove the correctness of Algorithm F, the primary subroutine in the compaction algorithm. Some of the proofs in this section use the correspondence between sketches and designs discussed in Chapter 8. To avoid confusion between sketches and sheets, we use the symbol Σ in place of S to denote a sketch.

Displacing traces and gates

One consequence of our definition of the sketch $\Sigma(\mathbf{d})$ is that gates and gate crossings transform nicely from Σ to $\Sigma(\mathbf{d})$. Let Γ be a partition of Σ, and for each gate $\gamma = p \triangleright q \in \Gamma$ define $\gamma(\mathbf{d})$ to be the linear path $p(\mathbf{d}) \triangleright q(\mathbf{d})$. Then $\gamma(\mathbf{d})$ is a horizontal cut of $\Sigma(\mathbf{d})$, and the homeomorphism $H(\cdot, \mathbf{d})$ that takes Σ to $\Sigma(\mathbf{d})$ maps $Im\,\gamma$ onto $Im\,\gamma(\mathbf{d})$. Hence the set $\Gamma(\mathbf{d}) = \{\,\gamma(\mathbf{d}) : \gamma \in \Gamma\,\}$ is a partition of $\Sigma(\mathbf{d})$. Furthermore, the restriction of $H(\cdot, \mathbf{d})$ to $Im\,\gamma$ is monotonic, because if L is the line containing $Im\,\gamma$, then $H(\cdot, \mathbf{d})\colon L \rightarrow L$ is a homeomorphism. Hence the intersections of $\gamma(\mathbf{d})$ by traces in $\Sigma(\mathbf{d})$ occur in the same order as the intersections of γ by traces in Σ. In other words, the intersection graphs of Σ with Γ and $\Sigma(\mathbf{d})$

with $\Gamma(\mathrm{d})$ are isomorphic. The isomorphism between the two graphs takes each island P to $P(\mathrm{d})$, each trace θ to $\theta(\mathrm{d})$, and so on.

The adjacency graph of a sketch is derived from its intersection graph by purely graph-theoretic operations. Hence two sketches with isomorphic intersection graphs also have isomorphic adjacency graphs.

Lemma 9d.1. *The adjacency graph of a sketch is independent of configuration.* □

One other fact about adjacency graphs is most conveniently proved within the sketch model.

Lemma 9d.2. *The skeleton of an adjacency graph is a forest.*

Proof. Let G be the adjacency graph and T its skeleton. Recall that T is obtained from G by deleting all gate arcs. Supposing that T contains a simple cycle C, we derive a contradiction. Let λ be a simple loop that passes through the regions forming C, making one crossing with each trace that corresponds to an arc in C, but no crossings with gates. Because λ enters no region twice, none of these traces can intersect λ more than once. Hence they all end inside λ, which means $\mathit{inside}(\lambda)$ contains a feature. Let X denote the set of points in the routing region that lie on no gate. By Corollary 2c.6, λ is essential in X. But since the gates form a partition of the routing region, the components of X are simply connected. This contradiction tells us that T is a forest. □

Structure of the adjacency graph

To understand the adjacency graph, we pass to the design model as in Chapter 8. Suppose $\Sigma = (\Xi, \Theta)$ is a sketch with partition Γ. Let S_ϵ be the corresponding sheet with design Θ^b, and let Γ^b denote the set $\{\, \gamma^b : \gamma \in \Gamma \,\}$. Let ϵ be small enough that every path in Γ^b is a link in S_ϵ. Then Γ^b is a pattern for S_ϵ because Γ is a partition of Σ. In addition, the cuts in Γ^b are disjoint. Now let ϵ be small enough that every interval of overlap between cuts in Γ and traces in Θ, except those where the trace does not cross over the cut, lies within $S_\epsilon - \mathit{Bd}\, S_\epsilon$. Then the intersection graph of Γ and Σ is also the graph of intersections between Γ^b and Θ^b in S_ϵ.

We interpret the construction of the reduced intersection graph of Σ as the construction of an embedding of Θ^b that conforms with Γ^b. See Section 7B for a description of the latter process. The intersection graph ignores overlaps where traces fail to cross over gates; ignoring these corresponds to making Θ^b stable with respect to Γ^b. To reduce the intersection graph, we find two nodes that are adjacent via both a trace and a gate, and remove whichever of those nodes represent crossings of the gate by the trace. I call the trace edge in this situation **removable**.

The process of removing a removable edge corresponds exactly to the collapsing of a collapsible subpath of a wire in $\Theta^\flat$. A subpath of a wire is collapsible when

(1) it connects two crossings or connects a crossing to a terminal,

(2) the things it connects are also connected by a subpath of a seam,

(3) the middle of the wire subpath crosses no other seams,

(4) the middle of the seam subpath crosses no other wires, and

(5) the two subpaths are path-homotopic or form a trivial link.

Conditions (1) through (4) say that the wire subpath and the seam subpath correspond to arcs of the intersection graph that connect the same nodes (terminals or crossings). Next we show that condition (5) is superfluous, completing the correspondence between removable edges and collapsible subpaths. One can check that the effects on the intersection graph of removing a removable edge and of collapsing a collapsible subpath are identical.

The homotopy condition (5) is a consequence of the others because $\Gamma^\flat$ is a pattern. Suppose $\omega_{s:t}$ and $\gamma_{a:b}$ are the wire and seam subpaths, satisfying conditions (1) through (4), where $\omega \in \Theta^\flat$ and $\gamma \in \Gamma^\flat$. There are two cases: either (a,s) and (b,t) are both crossings, or else only (a,s) is a crossing and $\omega(t)$ shares a fringe F with $\gamma(b)$. In the former case, $\omega_{s:t}$ and $\gamma_{a:b}$ lie within a single piece of the pattern $\Gamma^\flat$ (because $\omega_{s:t}$ is clean in $\Gamma^\flat$), and hence they are path-homotopic. In the latter case, we must show that the link $\omega_{t:s} \star \gamma_{a:b}$ is trivial. Because terminals in Σ are points, and F is derived from such a terminal, there are at most two seams in $\Gamma^\flat$ incident on F. Hence there is a path κ in F from $\gamma(b)$ to $\omega(t)$ crosses over no seams, and leaves γ on the same side that $\omega_{s:t}$ does. Consequently the loop $\omega_{t:s} \star \gamma_{a:b} \star \kappa$ lies within a single piece of $\Gamma^\flat$, which makes it inessential. Hence the link $\omega_{t:s} \star \gamma_{a:b}$ is path-homotopic to $\widehat{\kappa}$ and therefore trivial.

Our conclusion is that the reduced intersection graph of Γ and Σ is the graph of intersections of $\Gamma^\flat$ and an embedding $\Phi^\flat$ of $\Theta^\flat$ having no collapsible subpaths. By Proposition 7b.7, this design $\Phi^\flat$ conforms with $\Gamma^\flat$. One important consequence is the following.

Lemma 9d.3. *Let T be the skeleton of an adjacency graph. Let b and b' be two nodes of T bordering a gate γ from below, and let a and a' be adjacent to b and b', respectively, across γ. The distance from a to a' in T is equal to the distance from b to b' in T.*

Proof. The dual of the adjacency graph is the reduced intersection graph of the pattern Γ and the sketch Σ. By construction, wherever a trace in this realization touches a gate in Γ, it crosses over that gate. Hence the nodes bordering γ from above and below are connected in the structure shown in Figure 9d-1. It suffices to show that no two nodes on the top row in Figure 9d-1 are identical, and similarly for the bottom row. If so, then the upper and lower paths in Figure 9d-1 are simple, and since T is a forest (Lemma 9d.2), a simple path between two nodes in T has minimal length. Finally, the sequence of traces crossed by the upper path is identical to that crossed by the lower path. The lemma will follow immediately.

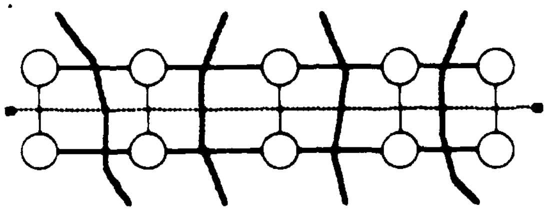


Figure 9d-1. *Nodes of the adjacency graph that border on a gate.* The striped line is the gate and the grey paths are parts of traces. Circles are nodes, light lines are gate edges, and dark lines are trace edges.

By symmetry, it is enough to prove that no two nodes on the top row are the same. Suppose otherwise; let N be a node of T that borders γ along two separate intervals. Every trace crossing over γ between these two intervals must turn and cross back without crossing any other gates in between. (We are using Theorem 2c.1 here.) Passing to the design $\Theta^\flat$ and the pattern $\Gamma^\flat$, some wire $\omega \in \Theta^\flat$ makes crossings (a, s) and (b, t) with a seam $\gamma^\flat$ such that the subpath $\omega_{s:t}$ is clean in $\Gamma^\flat$. Consequently $\omega_{s:t}$ and $\gamma_{a:b}$ lie within a single piece of $\Gamma^\flat$, and thus are path-homotopic. Therefore the crossings (a, s) and (b, t) of γ by ω are similar, contradicting the conformity of $\Theta^\flat$ with $\Gamma^\flat$. □

Use of the adjacency graph

Now we apply Corollary 7c.9 to characterize the flow across a cut in terms of the adjacency graph.

Proposition 9d.4. *The flow across a direct cut α in the sketch Σ is the length of the shortest path in the adjacency graph of Σ that*

(1) *begins at a node bordering on $\alpha(0)$ in the direction of α,*

(2) *ends at a node bordering on $\alpha(1)$ in the direction of $\widehat{\alpha}$, and*

(3) *has gate list equal to that of α.* □

Proof. Let G be the adjacency graph and Γ the partition of Σ used in its construction. We pass to the design model and consider the cut $\alpha^\flat$ derived from α. By Proposition 8a.5, the congestion of $\alpha^\flat$ settles at that of α, and since $\alpha^\flat$ is simple, its congestion equals its flow by Proposition 4b.6. So it suffices to show that the length of the shortest path in G satisfying (1)–(3) is the value at which $\mathit{flow}(\alpha^\flat, \Theta^\flat)$ settles.

We move everything into the setting of designs. The dual of G is the intersection graph of the pattern Γ^ϵ with the design Θ^ϵ derived from Σ for sufficiently small ϵ. Let $\theta_1, \ldots, \theta_m$ be a sequence of traces chosen from Θ, and let [illegible] be the gate list of α in Γ. Make ϵ small enough that the seam list of α [illegible] γ_n. There is a path in G which has properties [illegible] 3) and whose [illegible] $\theta_1, \ldots, \theta_m$ if and only if there is a link β^ϵ in Σ free in Γ^ϵ which:

(1) begins on the border containing $\alpha^\epsilon(0)$,

(2) ends on the border containing $\alpha^\epsilon(1)$,

(3) has seam list equal to that of α^ϵ,

and has wire list $\theta^\epsilon_1, \ldots, \theta^\epsilon_m$ in Θ^ϵ. Conditions [illegible] and (2) [illegible] requirement that the link β^ϵ have the same roots in Γ^ϵ as α^ϵ has.

One can reduce to the case in which α^ϵ conforms with [illegible] Corollary 7.[illegible] then completes the proof. It says that flow(α^ϵ, Θ^ϵ) equals [illegible] of the shortest wire list of a link β^ϵ that is free in Γ^ϵ and has the same [illegible] seam list in Γ^ϵ as α^ϵ has. □

Correctness of Algorithm F

The correctness of Algorithm F follows from [illegible] Proposition 9d.4.

Proposition 9d.5. *Let γ be a* [illegible] *of Σ, and let* [illegible] *γ_n be the gate list of γ. Then Algorithm F* [illegible] *γ and ϵ returns flow γ.*

Proof. Let x be a node of G bordering on $\gamma(0)$ [illegible] be a node of G bordering on $\gamma(1)$ in the direction of γ. [illegible] in Algorithm F computes the length of some path with gate [illegible] from x to a node bordering ψ. All such paths have length [illegible] Proposition 9d.4. So it suffices to show that DIST(x, ψ) [illegible] flow(γ) [illegible]. By Proposition 9d.4, there exists a path π in G that begins at a [illegible] on $\gamma(0)$ in the direction of γ, ends at a node y bordering on γ [illegible] of γ, and has gate list $\gamma_1, \ldots, \gamma_n$ and length flow(γ). Let π_0 be the [illegible] this path up to gate γ_1; for $0 < i < n$, let π_i be the portion of π between gates γ_i and γ_{i+1}; let π_n be the portion of π beyond gate γ_n.

We consider only the execution of DIST on x and ψ. Define [illegible] values of t and u just after iteration i of the loop in lines 4–7 [illegible]. Denote the length of a path α in G by $\ell(\alpha)$. Algorithm F maintains the following invariant.

There is a path α_i in T from u_i to the origin of π_i such that $t_i + \ell(\alpha_i) \le \sum$ [illegible]

[illegible] after the loop completes, we have $\ell(\alpha_n) + t_n \leq \sum_{j=0}^{n-1} \ell(\pi_j)$. Let T be [illegible] The concatenation of α_n and π_n, written $\alpha_n \cdot \pi_n$, is a path in T [illegible] Proposition 9d 4,

$$
\begin{aligned}
\mathrm{DIST}(x, y) &\leq t_n + \ell(\alpha_n \cdot \pi_n) \\
&= (\ell(\alpha_n) + t_n) + \ell(\pi_n) \\
&\leq (\textstyle\sum_{j=0}^{n-1} \ell(\pi_j)) + \ell(\pi_n) \\
&= \ell(\pi) = \mathit{flow}(\gamma).
\end{aligned}
$$

[illegible] the lemma

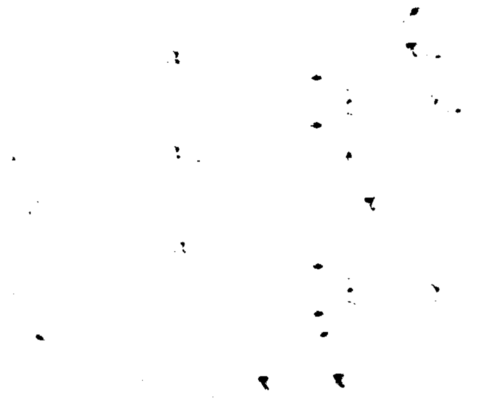


Figure 9d 2. *The inductive step in proving the correctness of Algorithm F.* A minimal path from x to y is $\pi_0 \cdots \pi_n$ (on the right), and the greedy path found by Algorithm F is $\rho_0 \cdots \rho_n$ (on the left). The induction hypothesis is that $\ell(\rho_0 \cdots \rho_{i-1}) + \ell(\alpha) \leq \ell(\pi_0 \cdots \pi_{i-1})$.

[illegible] the invariant, which we do by induction on i. The basis [illegible] Now assuming the invariant for i, we prove it for $i + 1$. See [illegible] a shortest path in T from u_i to a node bordering [illegible] are the shortest paths in T between [illegible] forest, by Lemma 9d 2, the shortest path between [illegible] path between them. In particular, α'_{i+1} is the [illegible] endpoints. The nodes adjacent to γ_{i+1} from below [illegible] along α_{i+1} is adjacent to γ_{i+1} from below. [illegible] path, the shortest path between u_i and the end [illegible]

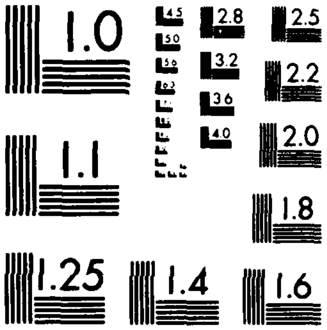
$$[illegible] \ell(\alpha_i) + \ell(\pi_i)$$

[illegible] induction hypothesis, we obtain

$$[illegible] + \ell(\alpha_{i+1}) \leq \sum_{j=0}^{i} \ell(\pi_j).$$

AD-A186 990 SINGLE-LAYER WIRE ROUTING(U) MASSACHUSETTS INST OF TECH 4/4
CAMBRIDGE LAB FOR COMPUTER SCIENCE F M MALEY AUG 87
MIT/LCS/TR-403 N00014-80-C-0622

UNCLASSIFIED F/G 9/1 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

The first term on the right is just t_{i+1}, and Lemma 9d.3 shows that $\ell(\alpha'_{i+1}) = \ell(\alpha_{i+1})$. We conclude that the invariant holds for $i + 1$. □

9E. The Abstract Compaction Algorithm

To prove the correctness of Algorithm C, the compaction algorithm, we proceed by way of an intermediate procedure called Algorithm A, the **abstract** compaction algorithm. The name derives from the fact that Algorithm A (which is not really an algorithm at all, but just a mathematical definition) abstracts the essential element of Algorithm C, namely the iterative definition of the subspace of configurations to be searched for a minimum width sketch. Algorithm A defines a sequence A_0, A_1, …, A_m of increasingly restricted subsets of the configuration space. These sets will correspond to sets of configurations satisfying the constraint system I at different stages of Algorithm C.

The first part of this section is devoted to the statement of Algorithm A and its preconditions. The rest of the section demonstrates the correctness of Algorithm A by proving the following theorem.

Theorem 9e.1. *The output A_m of Algorithm A is the connected component of* $\{\, \mathbf{c} \in \mathrm{C}(S) : S(\mathbf{c}) \text{ is routable} \,\}$ *that contains the initial configuration* $\mathbf{0}$.

Section 9F will draw a correspondence between Algorithms C and A, and prove that A_m is precisely the set of configurations that satisfy the final constraint system I of Algorithm C. Together with Theorem 9e.1, this implies that the constraints generated by Algorithm C are both necessary and optimal, if only convex constraints are allowed. Finally, because Algorithm C finds an optimal configuration among those satisfying the constraint system, it will follow that Algorithm C is correct, and that it finds the best solution available to any algorithm of its type.

There are at least two reasons for taking this abstract approach. First of all, it simplifies the correctness proof by separating the mathematical from the algorithmic concerns. Second, and more important, it clarifies the assumptions on which the compaction algorithm relies. An understanding of these assumptions will allow Algorithm C to be easily modified.

Assumptions

The input to Algorithm A is a routable sketch S together with a sequence $\Psi(S) = \langle \psi_1, \ldots, \psi_m \rangle$ of potential cuts of S; the output is a set of configurations A_m. As a precondition of Algorithm A, the potential cuts $\Psi(S)$ must determine the routability of the modified sketches $S(\mathbf{d})$. Specifically, they must have the following property.

Routability property. Assume $S(\mathbf{0})$ is routable, and let $\mathbf{d} \in \mathbf{C}(S)$ be a configuration. If $\mathbf{d}$ fails to protect some element of Ψ, then $S(\mathbf{d})$ is not routable. But if for all $t \in [0,1]$ the configuration $t\mathbf{d}$ protects every $\psi \in \Psi(S)$, then $S(\mathbf{d})$ is routable.

The capacities of the potential cuts must also have a special property.

Convexity property. For each $\psi \in \Psi(S)$, the function $\mathbf{d} \mapsto cap(\psi(\mathbf{d}))$ is convex.

Actually a weaker property suffices, namely that for each line L in configuration space, there is a point $\mathbf{c}$ of L at which the capacity $cap(\psi(\mathbf{c}))$ is minimal, and $cap(\psi(\mathbf{d}))$ is nondecreasing as $\mathbf{d}$ moves away from $\mathbf{c}$ along L. The simpler condition of convexity is general enough for my purposes, however.

In principle, my compaction method depends on only two further assumptions about the potential cuts $\Psi(S)$.

Ordering property. Let the configuration $\mathbf{d} \in \mathbf{C}(S)$ protect ψ_i for all $i < k$. If $\mathbf{d}$ lies on the frontier of $\{\, \mathbf{c} \in \mathbf{C}(S) : \psi(\mathbf{c}) \text{ is a cut} \,\}$, then every cut that is a subpath of $\psi_k(\mathbf{d})$ is either safe or empty.

Boundary property. The configuration space $\mathbf{C}(S)$ is open in R^n, and there is a closed set $D \subseteq R^n$ such that all configurations in $\mathbf{C}(S) - D$ fail to protect some potential cut in Ψ.

In practice, of course, we also desire that the sequence $\langle \psi_i \rangle$ be computable in polynomial time. As we show in Section 9F, the sequence of potential cuts examined by Algorithm C has all these desirable properties.

The abstract algorithm

Before plunging into the algorithm, I shall provide a brief overview. Algorithm A computes a sequence of polytopes in configuration space, each one contained in the last. The configurations in the kth polytope will protect the first k potential cuts in $\Psi(S)$. To process ψ_k, the kth potential cut, the algorithm first determines whether ψ_k is unsafe and nonempty in any configuration in the current polytope. If not, the algorithm ignores ψ_k. Otherwise, it defines a set of unacceptable configurations in which the capacity of ψ_k falls below some critical value. This set contains all configurations in the current polytope that fail to protect ψ_k. Its complement consists of two half-spaces: one in which the lower endpoint of ψ_k is far to the right of the upper endpoint, and one in which the situation is reversed. Because the initial configuration is always acceptable, it must fall into one half-space or the other; the kth polytope is determined by intersecting the $(k-1)$st polytope with the half-space that contains $\mathbf{0}$. Thus Algorithm A eliminates configurations not reachable from the initial one.

Algorithm A. (Finds the set of acceptable modifications of a sketch.)
Input: a legal sketch S with n modules specified, and a sequence $\langle \psi_1, \ldots, \psi_m \rangle$ of potential cuts of S with the routability, convexity, ordering, and boundary properties.
Output: a subset of the configuration space $\mathbf{C}(S)$.
Local variables: an integer k, polytopes A_k of acceptable configurations, sets U_k of unacceptable configurations, and inequalities Λ_k.

```
1. $A_0 \leftarrow \mathbf{C}(S)$;
2. for $k \leftarrow 1$ to $m$ do
3.   if some $\mathbf{c} \in A_{k-1}$ does not protect $\psi_k$ then
4.     $U_k \leftarrow \{\, \mathbf{d} \in R^n : cap(\psi_k(\mathbf{d})) < flow(\psi_k(\mathbf{c})) \,\}$;
5.     If the endpoints of $\psi_k$ lie on the features $P_k$ and $Q_k$, write $U_k$ as
           $\{\, \mathbf{d} : \Delta^- < \Delta_{P_k Q_k}(\mathbf{d}) < \Delta^+ \,\}$. Either $0 \in (-\infty, \Delta^-]$ or $0 \in [\Delta^+, \infty)$.
6.     $\Lambda_k \leftarrow \begin{cases} \Delta_{P_k Q_k}(\mathbf{d}) \geq \Delta^+, & \text{if } 0 \geq \Delta^+; \\ \Delta_{P_k Q_k}(\mathbf{d}) \leq \Delta^-, & \text{if } 0 \leq \Delta^-; \end{cases}$
7.     $A_k \leftarrow \{\, \mathbf{d} \in A_{k-1} : \mathbf{d}$ satisfies $\Lambda_k \,\}$
8.   else $A_k \leftarrow A_{k-1}$; $U_k \leftarrow \emptyset$;
9. return $A_m$.
```

Some remarks about Algorithm A are in order.

- The set U_k is defined in terms of an arbitrary configuration $\mathbf{c} \in A_{k-1}$ that fails to protect ψ_k. We will soon show that U_k is independent of the choice of $\mathbf{c}$.
- The constraint Λ_k is a simple linear inequality between $d_{\mu(P_k)}$ and $d_{\mu(Q_k)}$, and hence defines a closed half-space in R^n. Since A_0 is convex, the set A_k is therefore convex for each k.
- In the light of the following results, the definition of A_k in lines 6–7 may be read as "A_k is the component of $A_{k-1} - U_k$ that contains $\mathbf{0}$".

Core of the correctness proof

The following definition and lemma are fundamental to the correctness proof. The lemma's proof reveals the purpose of the convexity and ordering properties.

Definition 9e.2. Two configurations, $\mathbf{d}$ and $\mathbf{d}'$, are **equivalent** with respect to a potential cut ψ if for every configuration $\mathbf{b}_t = (1-t)\mathbf{d} + t\mathbf{d}'$ with $t \in [0,1]$ the path $\psi(\mathbf{b}_t)$ is a cut. This relation is written '$\mathbf{d} \approx \mathbf{d}'$ with respect to ψ'.

In configurations that are equivalent with respect to a potential cut ψ, the flow across ψ is equal. To see why, suppose $\mathbf{d} \approx \mathbf{d}'$ with respect to ψ, and let

$H: R^2 \times \mathbf{C}(S) \to R^2$ be the map used to define the sketch $S(\mathbf{c})$ for $\mathbf{c} \in \mathbf{C}(S)$. (We have $S(\mathbf{c}) = H(\cdot, \mathbf{c}) \diamond S(\mathbf{0})$.) Because $H(\cdot, \mathbf{d})$ and $H(\cdot, \mathbf{d}')$ are homeomorphisms, we have

$$\text{flow across } \psi(\mathbf{d}) \text{ in } S(\mathbf{d}) = \text{flow across } H(\cdot, \mathbf{d})^{-1} \circ \psi(\mathbf{d}) \text{ in } S(\mathbf{0}), \text{ and}$$
$$\text{flow across } \psi(\mathbf{d}') \text{ in } S(\mathbf{d}') = \text{flow across } H(\cdot, \mathbf{d}')^{-1} \circ \psi(\mathbf{d}') \text{ in } S(\mathbf{0}).$$

A bridge homotopy between the two cuts on the right is $(s,t) \mapsto \theta_t(s)$ where $\theta_t = H(\cdot, \mathbf{b}_t)^{-1} \circ \psi(\mathbf{b}_t)$. (Both ψ and H are piecewise linear.) Hence they have the same flow (congestion) in $S(\mathbf{0})$ by Corollary 8a.6. For similar reasons, either $\psi(\mathbf{d})$ and $\psi(\mathbf{d}')$ are both empty or else both are nonempty.

Lemma 9e.3. *Let ψ be a potential cut in the sketch S, let* $\mathbf{d}$ *and* $\mathbf{d}'$ *be configurations in $\mathbf{C}(S)$, and put $\mathbf{b}_t = (1-t)\mathbf{d} + t\mathbf{d}'$ for $t \in [0,1]$. Suppose that the capacity function $t \mapsto cap(\psi(\mathbf{b}_t))$ is convex, and that whenever $\mathbf{b}_t$ lies on the frontier of $\{\, \mathbf{c} \in \mathbf{C}(S) : \psi(\mathbf{c}) \text{ is a cut} \,\}$, all cuts that are subpaths of $\psi(\mathbf{b})$ are safe or empty.*

(1) *If* $\mathbf{d}'$ *protects ψ but* $\mathbf{d}$ *does not, then $cap(\psi(\mathbf{d}')) \geq flow(\psi(\mathbf{d}))$.*

(2) *If neither* $\mathbf{d}$ *nor* $\mathbf{d}'$ *protects ψ, then $\mathbf{d} \approx \mathbf{d}'$ with respect to ψ.*

Proof. As t varies from 0 to 1, the sketch $S(\mathbf{b}_t)$ varies from $S(\mathbf{d})$ to $S(\mathbf{d}')$, and the linear path $\sigma_t = \psi(\mathbf{b}_t)$ is sometimes a cut, and sometimes it crosses features. Denote the flow across σ_t by $f_t = flow(\psi(\mathbf{b}_t))$, and the capacity (or "length") of σ_t by $l_t = cap(\psi(\mathbf{b}_t))$.

We first argue that the set $Z = \{\, t \in [0,1] : \sigma_t \text{ is a cut} \,\}$, considered as a subspace of the unit interval, is open. Let σ_t be a cut; say it connects the features P and Q. There is some positive distance between σ_t and every feature but P and Q; because $\mathbf{b}_t \in \mathbf{C}(S)$, no other features can touch the endpoints of σ_t. And since σ_t and the module positions in $S(\mathbf{b}_t)$ are all continuous functions of t, there is some neighborhood U of t such that σ_u is a cut whenever $u \in U$. So Z is open, and hence it consists of disjoint intervals, each one open in $[0,1]$.

Now let us focus attention on one of these intervals, call it T. For all $s, t \in T$ the configurations $\mathbf{b}_s$ and $\mathbf{b}_t$ are equivalent with respect to ψ. Hence the flow f_t is a constant f_T for all $t \in T$. And if s lies on the frontier of T, considering T as a subspace of I, then σ_s is not a cut. The following claim is the crux of the argument.

Claim: *If $t \in T$ and $s \in Fr\, T$, the configuration $\mathbf{b}_t$ protects ψ unless $l_t < l_s$.*

Consider the sketch $S(\mathbf{b}_s)$. At this point, one or more features have just contacted σ_s, and hence σ_s is broken up into a sequence of cuts $\alpha_1, \ldots, \alpha_l$. Because $\mathbf{b}_s$ is on the frontier of the set of configurations that make ψ a cut, all the cuts α_i are safe or empty. If the cut σ_t is empty, then ψ is fixed with respect to the module that contains its endpoints, and so the cuts α_1 and α_l must connect different modules. Thus α_1 and α_l are safe, not empty; their capacities are nonnegative.

Hence $l_s \geq 0$ also. In this case $f_T \leq l_s$ because $f_T = 0$. Now suppose that σ_t is not empty. If f_T were to exceed l_s, one of these cuts α_i would be unsafe and nonempty. One could prove this rigorously by passing to the design model and appealing to Proposition 4f.1 and Lemma 4f.3. We conclude that $f_T \leq l_s$. If $\mathbf{b}_t$ fails to protect ψ then $l_t < f_T$, so $l_t < l_s$. This proves the claim.

The lemma is now straightforward. Both parts of the lemma assume that $\mathbf{d}$ fails to protect ψ, so we may assume that $\sigma_0 = \psi(\mathbf{d})$ is a cut, and that $l_0 < f_0$. Suppose first that $\mathbf{d}$ and $\mathbf{d}'$ are equivalent with respect to ψ. Then $f_1 = f_0$, and neither σ_0 nor σ_1 is empty. If $\mathbf{d}'$ protects ψ, then σ_1 is safe, and so $l_1 \geq f_1$. Thus $l_1 \geq f_0$, establishing (1). Conclusion (2) is trivial if $\mathbf{d} \approx \mathbf{d}'$, so we now assume $\mathbf{d} \not\approx \mathbf{d}'$ with respect to ψ. Then there exists $t \in (0,1]$ such that σ_t is not a cut. Let s be the smallest such value, and consider the interval $T = [0, s)$. Since $\mathbf{d} = \mathbf{b}_0$ does not protect ψ, the claim implies $l_0 < l_s$. Now because the function $t \mapsto l_t$ is convex, it has at most one local minimum in $[0,1]$. Because $l_0 < l_s$, the minimum value of l_t must occur in the interval $(-\infty, s)$. Hence l_t is nondecreasing on $[s, 1]$, and we have $l_1 \geq l_s \geq f_0$. This proves conclusion (1), because l_1 is $cap(\psi(\mathbf{d}'))$ and f_0 is $flow(\psi(\mathbf{d}))$. Now we prove (2) by showing that $\mathbf{d}'$ protects ψ. If σ_1 is a cut, let β be the largest value such that σ_β is not a cut. (One must exist, for we are assuming $\mathbf{d} \not\approx \mathbf{d}'$.) Applying the claim to the interval $T = (\beta, 1]$, we find that $\mathbf{b}_1$ protects ψ because $l_1 \geq l_\beta$. Since $\mathbf{b}_1 = \mathbf{d}'$, this proves statement (2). □

Body of the correctness proof

Lemma 9e.3 provides us with the following lemma, our main tool for proving Theorem 9e.1. We shall use this lemma frequently.

Lemma 9e.4. (Potential Cut Lemma) *Suppose $1 \leq k \leq m$, and let* $\mathbf{d}$ *and* $\mathbf{d}'$ *be configurations in A_{k-1}.*

(1) *If* $\mathbf{d}'$ *protects ψ_k but* $\mathbf{d}$ *does not, then $cap(\psi_k(\mathbf{d}')) \geq flow(\psi_k(\mathbf{d}))$.*

(2) *If neither* $\mathbf{d}$ *nor* $\mathbf{d}'$ *protects ψ_k, then $\mathbf{d} \approx \mathbf{d}'$ with respect to ψ_k.*

Statement (2) implies that any two configurations $\mathbf{d}, \mathbf{d}' \in A_{k-1}$ that fail to protect ψ_k must satisfy $flow(\psi_k(\mathbf{d})) = flow(\psi_k(\mathbf{d}'))$. Thus Lemma 9e.4 shows that the sets U_k defined in line 4 of Algorithm A are uniquely determined.

The proof of Lemma 9e.4 depends on several facts about the set A_{k-1}. In particular, the lemma makes no sense unless A_{k-1} is well defined. On the other hand, A_k is well defined only if the Potential Cut Lemma holds for A_{k-1}. We must therefore prove Lemma 9e.4 in parallel with the following claim.

Lemma 9e.5. *For $1 \leq k \leq m$, the following statements hold:*

(3) *If $\mu(P_k) = \mu(Q_k)$, then every configuration* $\mathbf{c} \in A_{k-1}$ *protects ψ_k.*

(4) *The set A_k is well defined by Algorithm A.*
(5) *The point* **0** *lies in A_k.*
(6) *Every configuration in A_k protects the potential cuts ψ_1 through ψ_k.*

Proof of Lemmas 9e.4 and 9e.5. The proof proceeds by induction on k, with the inductive hypothesis being the conjunction of (4) through (6). A basis for this hypothesis is easily established at $k = 0$: the set A_0 is obviously well defined, $\mathbf{0} \in A_0$ by definition, and condition (6) is vacuously true. So assume $k \geq 1$. The key step is the proof of (1) and (2), in Lemma 9e.4, from the inductive hypothesis.

(1,2) We apply Lemma 9e.3 to the configurations $\mathbf{d}$ and $\mathbf{d}'$ and the potential cut ψ_k. The convexity property implies that the function $\mathbf{b} \mapsto cap(\psi(\mathbf{b}))$ is convex, and hence $t \mapsto cap(\psi(\mathbf{b}_t))$ is convex. And since A_{k-1} is a convex set, the inductive hypothesis implies that every configuration $\mathbf{c} \in L$ protects the potential cuts ψ_1 through ψ_{k-1}. This fact, combined with the ordering property, demonstrates the final assumption of Lemma 9e.3. The conclusion of that lemma is identical to the conclusion of Lemma 9e.4.

(3) Suppose $\mu(P_k) = \mu(Q_k)$, and apply parts (1) and (2) to ψ_k with $\mathbf{0}$ in place of $\mathbf{d}'$ and $\mathbf{c}$ in place of $\mathbf{d}$. Since $S(\mathbf{0})$ is routable, the routability property implies that $\mathbf{0}$ protects ψ_k. Hence only part (1) can apply; it says that $cap(\psi_k(\mathbf{0})) \geq flow(\psi_k(\mathbf{c}))$ if $\mathbf{c}$ fails to protect ψ_k. But our assumption that $\mu(P_k) = \mu(Q_k)$ implies that the capacity of ψ_k is independent of configuration. Therefore $cap(\psi_k(\mathbf{c})) \geq flow(\psi_k(\mathbf{c}))$, and so $\psi_k(\mathbf{c})$ cannot be unsafe. Therefore $\mathbf{c}$ protects ψ_k.

(4) For A_k to be well defined, the set U_k defined in line 4 of Algorithm A must have the specific form $\{\, \mathbf{d} \in \mathbf{C}(S) : \Delta^- < \Delta_{P_kQ_k}(\mathbf{d}) < \Delta^+ \,\}$, for some Δ^- and Δ^+. Recall that U_k includes a point $\mathbf{d}$ if and only if the capacity $cap(\psi_k(\mathbf{d}))$ of $\psi_k(\mathbf{d})$ is less than the constant $f = flow(\psi_k(\mathbf{c}))$. But by the definition of a potential cut, $\psi_k(\mathbf{d})$ depends only on $\Delta_{P_kQ_k}(\mathbf{d})$. Hence it suffices to show that the set

$$\{\, \Delta_{P_kQ_k}(\mathbf{d}) : \mathbf{d} \in R^n \text{ and } cap(\psi_k(\mathbf{d})) < f \,\}$$

is a nonempty open interval (Δ^-, Δ^+). By part (3), line 4 is only reached if $\mu(P_k) \neq \mu(Q_k)$. Hence we may choose a line L through $\mathbf{c}$ on which $\Delta_{P_kQ_k}(\mathbf{d})$ is not constant. The convexity property of ψ_k implies that the set $\{\, \mathbf{d} \in L : cap(\psi_k(\mathbf{d})) < f \,\}$ is a open interval of L; it is nonempty because it contains $\mathbf{c}$. Since $\Delta P_kQ_k(\mathbf{d})$ is a nonconstant linear function on L, the set

$$\{\, \Delta_{P_kQ_k}(\mathbf{d}) : \mathbf{d} \in L \text{ and } cap(\psi_k(\mathbf{d})) < f \,\}$$

is also a nonempty open interval. This is enough, because every value $\Delta_{P_kQ_k}(\mathbf{d})$ is represented by some $\mathbf{d} \in L$.

(5) By the induction hypothesis, $\mathbf{0} \in A_{k-1}$. If every $\mathbf{c} \in A_{k-1}$ protects ψ_k, then $\mathbf{0} \in A_k$ trivially. Otherwise, apply (1) to $\mathbf{0}$ and $\mathbf{c}$. (Because $S(\mathbf{0})$ is routable, $\mathbf{0}$ protects ψ_k by the routability property.) So $\mathit{cap}(\psi_k(\mathbf{0})) \geq \mathit{flow}(\psi_k(\mathbf{c}))$, whence $\mathbf{0} \notin U_k$. Because $\Delta_{P_kQ_k}(\mathbf{0}) = 0$, by definition, we have $0 \notin (\Delta^-, \Delta^+)$. Thus $\mathbf{0}$ satisfies the constraint Λ_k defined at line 6, and so $\mathbf{0} \in A_k$.

(6) Since $A_k \subseteq A_{k-1}$, every configuration $\mathbf{d} \in A_k$ protects ψ_1 through ψ_{k-1}, by the induction hypothesis; it remains to show that every configuration $\mathbf{d} \in A_k$ protects ψ_k. Suppose that $\mathbf{d} \in A_{k-1}$ fails to protect ψ_k. Then U_k is nonempty, and is defined in terms of some configuration $\mathbf{c}$. By part (2), $\mathbf{d} \approx \mathbf{c}$ with respect to ψ_k, and in particular $\mathit{flow}(\psi_k(\mathbf{d})) = \mathit{flow}(\psi_k(\mathbf{c}))$. Because $\mathbf{d}$ does not protect ψ_k, certainly $\mathit{cap}(\psi_k(\mathbf{d})) < \mathit{flow}(\psi_k(\mathbf{d}))$, and it follows that $\mathbf{d} \in U_k$. But the constraint Λ_k excludes all members of U_k from A_k. Therefore $\mathbf{d} \notin A_k$. □

From the above lemma, most of Theorem 9e.1 follows quickly. First of all, the initial configuration $\mathbf{0}$ is a member of A_m by claim (5). Second, if $\mathbf{d} \in A_m$, then for all $t \in [0,1]$, the configuration $t\mathbf{d}$ lies in A_m, and hence protects every $\psi \in \Psi(S)$ by claim (6). Therefore by the routability property, $S(\mathbf{d})$ is routable for all $\mathbf{d} \in A_m$. It remains to argue that A_m is a single connected component of $\{\, \mathbf{d} \in \mathrm{C}(S) : S(\mathbf{d}) \text{ is routable} \,\}$. To do so, we make use of an elementary topological result. A subset X of a topological space is said to **surround** another subset Y if Y lies in the interior of X, and the closure of Y is contained in X. If X surrounds the nonempty set Y, then Y is a connected component of the complement of $X - Y$.

Lemma 9e.6. *For $0 \leq k \leq m$, the set A_m is surrounded by the region*

$$X_k = A_k \cup \left(\bigcup_{i=1}^{k} A_{k-1} \cap U_k \right).$$

Proof. It suffices to show that A_m is closed and X_k is open, because clearly $A_m \subseteq X_k$. First the former. By the boundary property, the configurations that protect all the potential cuts $\Psi(S)$ lie within a closed subset D of $\mathrm{C}(S)$. By claim (6) of Lemma 9e.5, all points of A_m protect every $\psi \in \Psi(S)$. Therefore A_m is the intersection of D with the set of configurations that satisfy the inequalities Λ_k. Each configuration Λ_k defines a closed subset of R^n. Therefore A_m is closed.

Now we prove by induction on k that X_k is open. The basis case, $X_0 = A_0$, is guaranteed by the boundary property. Let $k > 0$, and consider the nontrivial case when U_k is nonempty. From the definition of X_k we derive $X_k = (X_{k-1} - A_{k-1}) \cup A_k \cup (A_{k-1} - U_k)$, which reduces to $X_{k-1} - (A_{k-1} - U_k - A_k)$. The set $B = A_{k-1} - U_k - A_k$ is the intersection of A_{k-1} with one of the closed half-spaces forming the complement of U_k; it remains to show that B is closed in X_{k-1}. But A_{k-1} is just the subset of X_{k-1} satisfying the constraints Λ_i, for all $i < k$, so B is

X_{k-1} intersected with finitely many closed half-spaces. Therefore $X_k = X_{k-1} - B$ is open. □

Setting $k = m$ in Lemma 9e.6, we find that $\bigcup_{i=1}^{m}(A_{i-1} \cap U_i)$ disconnects A_m from the rest of R^n. Hence the connected component of $\{ \mathbf{d} \in \mathbf{C}(S) : S(\mathbf{d}) \text{ is routable} \}$ that contains A_m cannot be a proper superset of A_m, unless it also contains a point in $A_{i-1} \cap U_i$ for some i. But if $\mathbf{d} \in A_{i-1}$ corresponds to a routable sketch, then (by the routability property) it protects ψ_i, and statement (1) of the Potential Cut Lemma applies to $\mathbf{d}$ and the configuration $\mathbf{c} \in A_{i-1}$ used to define U_i. It shows that $cap(\psi_i(\mathbf{d})) \geq flow(\psi_i(\mathbf{c}))$, which means that $\mathbf{d} \notin U_i$. Therefore $\mathbf{d} \in A_{i-1} \cap U_i$ implies that $S(\mathbf{d})$ is not routable. So A_m is precisely equal to the component of $\{ \mathbf{d} \in \mathbf{C}(S) : S(\mathbf{d}) \text{ is routable} \}$ that contains $\mathbf{0}$. This completes the proof of Theorem 9e.1.

9F. Implementing the Abstract Algorithm

In this section, we build upon the results of Sections 9D and 9E to prove the correctness of Algorithm C, the concrete compaction algorithm. The hard part of the proof is over: Algorithm A, which is an abstract description of the compaction algorithm, is proven correct by Theorem 9e.1 of the previous section. It remains to show that Algorithm C is just a special case of Algorithm A. There are two steps to this process: first, to identify the potential cuts that Algorithm C uses, and show that they satisfy the preconditions of Algorithm A; and second, to prove an explicit correspondence between the quantities computed by the two algorithms. The correctness of the compaction algorithm will then follow from the correctness of its subroutines (Algorithms F and R) along with Theorem 9e.1.

Preconditions of Algorithm A

Our first task is to show that the potential cuts used by Algorithm C satisfy the requirements of Algorithm A, namely the routability, capacity, ordering, and boundary properties. The potential cuts in question are of three types.

(1) Horizontal potential cuts ϕ_{pq} where either p or q is a feature endpoint.

(2) Diagonal potential cuts ϕ_{pq} where p and q are feature endpoints.

(3) Critical potential cuts χ_{pQ} where p is a feature endpoint.

Let S denote the sketch input to Algorithm C, and let $\Psi(S)$ contain all the potential cuts for S of types 1–3. We number these cuts $\psi_1, \ldots, \psi_m$ in the order that Algorithm C examines them. Since Algorithm C considers horizontal potential cuts first, the cuts of type (1) are $\psi_1, \ldots, \psi_h$ for some h. Next come the potential cuts of type 2 in order of height, and finally the potential cuts of type 3 in order of height.

We treat Algorithm C as if it processed all the potential cuts in $\Psi(S)$, although it actually ignores those that connect features in the same module. Part (3) of Lemma 9e.5 says that such potential cuts generate no constraints; hence Algorithm C is justified in ignoring them.

Proposition 9f.1. *The sequence $\Psi(S)$ has the routability, convexity, ordering, and boundary properties.*

Proof. The routability property is easiest. If a configuration $\mathbf{d} \in \mathrm{C}(S)$ fails to protect some potential cut $\psi \in \Psi(S)$, then $\psi(\mathbf{d})$ is unsafe and nonempty in the sketch $S(\mathbf{d})$. By Proposition 8b.3, then, $S(\mathbf{d})$ is unroutable. On the other hand, if every configuration $t\mathbf{d}$ with $t \in [0,1]$ protects every potential cut in $\Psi(S)$, then in particular $\mathbf{d}$ protects all the critical potential cuts of S. By Theorem 9c.2, the sketch $S(\mathbf{d})$ is therefore routable.

To check the convexity property for a potential cut $\psi \in \Psi(S)$, it is enough to show that the function $\mathbf{d} \mapsto \|\psi(\mathbf{d})\|$ is convex on $\mathrm{C}(S)$. Let $\mathbf{d}_0, \mathbf{d}_1$ be arbitrary configurations in $\mathrm{C}(S)$, and for $t \in [0,1]$ define $\mathbf{d}_t = (1-t)\mathbf{d}_0 + t\mathbf{d}_1$. Say ψ connects the features P and Q. For each t we have $\psi(\mathbf{d}_t)(0) = p_t(\mathbf{d}_t)$ and $\psi(\mathbf{d}_t)(1) = q_t(\mathbf{d}_t)$ for some $p_t \in P$ and $q_t \in Q$. Put $l_t = \|q_t(\mathbf{d}_t) - p_t(\mathbf{d}_t)\| = \|\psi(\mathbf{d}_t)\|$. We must show that $l_t \leq (1-t)l_0 + tl_1$. If $\psi = \phi_{pq}$ for some p and q, then $q_t = q$ and $p_t = p$ for all $t \in [0,1]$. Consequently the vector $q_t(\mathbf{d}_t) - p_t(\mathbf{d}_t)$ changes linearly with t, and so the convexity of $\|\cdot\|$ implies that $l_t \leq (1-t)l_0 + tl_1$. Now suppose $\psi = \chi_{pQ}$ for some feature Q and feature endpoint p. Then $p_t = p$ for all t, and q_t has the property that for all $q \in Q$,

$$\|q_t(\mathbf{d}_t) - p(\mathbf{d}_t)\| \geq \|q(\mathbf{d}_t) - p(\mathbf{d}_t)\|. \tag{9–2}$$

Because Q is a convex set, we may choose $q = (1-t)q_0 + tq_1$. Then $q(\mathbf{d}_t)$ is linear in t, and equals $q_0(\mathbf{d}_0)$ or $q_1(\mathbf{d}_1)$ if t is 0 or 1, respectively. Of course, $p(\mathbf{d}_t)$ is also linear in t. Hence by the convexity of $\|\cdot\|$, the right-hand side of (9–2) is at most $(1-t)l_0 + tl_1$. The left-hand side of (9–2) is just l_t, so the length of χ_{pQ} is a convex function. Therefore $\Psi(S)$ has the convexity property.

Now we argue that the sequence $\Psi(S) = \langle \psi_1, \ldots, \psi_m \rangle$ has the ordering property. Let the configuration $\mathbf{d} \in \mathrm{C}(S)$ protect ψ_i for all $i < k$, and suppose $\mathbf{d} \in Fr\{\, \mathbf{c} \in \mathrm{C}(S) : \psi(\mathbf{c}) \text{ is a cut} \,\}$. We show that every cut that is a subpath of $\psi_k(\mathbf{d})$ is either $\psi_i(\mathbf{d})$, for some $i < k$, or its reverse. Since $\mathbf{d}$ protects ψ_i, such cuts are either safe or empty. For $\mathbf{d}$ to lie in $Fr\{\, \mathbf{c} \in \mathrm{C}(S) : \psi_k(\mathbf{c}) \text{ is a cut} \,\}$ means that the features interrupting $\psi_k(\mathbf{d})$ must do so at their endpoints, and furthermore that $\psi_k(\mathbf{d})$ is not horizontal. If $\psi_k = \phi_{pq}$ for some feature endpoints p and q, then every cut that is a subpath of $\psi_k(\mathbf{d})$ begins and ends at feature endpoints, and has smaller height than ψ_k. All such cuts appear in the list $\langle \psi_1, \ldots, \psi_{k-1} \rangle$. The other case is only slightly harder. Suppose $\psi_k = \chi_{pQ}$ for some feature Q and feature endpoint p. Let

α be a subcut of $\psi_k(\mathbf{d})$ that ends on $Q(\mathbf{d})$, if one exists. Then all cuts that are subpaths of $\psi_k(\mathbf{d})$, except possibly α and $\widehat{\alpha}$, are cuts between feature endpoints; they have the form $\psi_i(\mathbf{d})$ for some $i < k$. If α exists, it is a critical cut from the feature endpoint $\alpha(0)$ to $Q(\mathbf{d})$, and has the form $\chi_{rQ}(\mathbf{d})$ where $\alpha(0) = r(\mathbf{d})$. As noted in Section 9C, the height of χ_{pQ} exceeds that of χ_{rQ}, and hence χ_{rQ} appears in $\langle \psi_1, \ldots, \psi_{k-1} \rangle$.

To check the boundary property, we must exhibit a closed set $D \subseteq R^n$ such that all configurations in $\mathbf{C}(S) - D$ fail to protect some potential cut in Ψ. (That $\mathbf{C}(S)$ is open follows directly from its definition.) Let w denote the minimum of the widths of the elements of S. The space $\mathbf{C}(S)$ was defined as the set of configurations $\mathbf{d}$ such that for all points p and q of S with $p_y = q_y$ and $p_x < q_x$, we have $\Delta_{pq}(\mathbf{d}) > 0$. We may assume that $\mu(p) \neq \mu(q)$. Define D the same way, but replace the condition $\Delta_{pq}(\mathbf{d}) > 0$ by the constraint $\Delta_{pq}(\mathbf{d}) \geq w$. Clearly D is closed in R^n. And if $\mathbf{d}$ is a configuration in $\mathbf{C}(\mathbf{d}) - D$, then there are two features in separate modules of $S(\mathbf{d})$ whose separation is less than w. Choose features P and Q such that the horizontal separation between $P(\mathbf{d})$ and $Q(\mathbf{d})$ is minimal. The minimum separation is realized at a feature endpoint, so there are points $p \in P$ and $q \in Q$ such that $\phi_{pq} \in \Psi(S)$ and $\|\phi_{pq}(\mathbf{d})\| < w$. By the choice of P and Q, no features intervene between $p(\mathbf{d})$ and $q(\mathbf{d})$, and hence $\phi_{pq}(\mathbf{d})$ is a cut. It is nonempty because it connects different islands, and is unsafe because its capacity is negative. Thus $\mathbf{d}$ fails to protect the potential cut $\phi_{pq} \in \Psi(S)$. □

Correspondence between the algorithms

The final phase of our proof strategy involves showing that the constraints computed by the concrete algorithm define the same space as the constraints Λ_k defined abstractly. This fact will imply that the compaction algorithm searches precisely the set A_m of acceptable configurations, and correctness will follow quickly. In order to state the correspondence, let C_0 denote the set of configurations satisfying the constraint system I defined at line 2 of Algorithm C, and let C_k denote those configurations satisfying I after the kth iteration on the loop in lines 3–6.

Lemma 9f.2. *For all k satisfying $h \leq k \leq m$, the sets C_{k-h} and A_k are identical.*

Proof. Recall that h is the number of horizontal cuts in the sequence $\Psi(S)$. We prove the lemma by induction on k, the basis case being $k = h$. Any configuration in A_h is in $\mathbf{C}(S)$, because $A_h \subseteq A_0$, and also protects the horizontal potential cuts, according to part (6) of Lemma 9e.5. Therefore $A_h \subseteq C_0$. On the other hand, you may check that when the constraint Λ_k exists, for $k \leq h$, it corresponds to the potential cut in I_0 induced by ψ_k. (Here we use Proposition 9d.5, which establishes the correctness of Algorithm F.) Therefore $C_0 \subseteq A_h$.

For the inductive step, suppose that $C_{k-h-1} = A_{k-1}$. We first draw a correspondence between the configurations $\mathbf{c}$ found by Algorithms A and C. The key observation is that the configuration $\mathbf{c}$ found by Algorithm C at line 4 minimizes the capacity $\mathit{cap}(\psi_k(\mathbf{c}))$ over all $t \in C_{k-h-1} = A_{k-1}$. (Dijkstra's algorithm is applicable here, because according to Lemma 9e.5, the initial configuration $\mathbf{0}$ satisfies the constraint system.) We wish to argue that if any $\mathbf{d} \in A_{k-1}$ fails to protect ψ_k, then neither does $\mathbf{c}$. Suppose to the contrary that $\mathbf{c}$ protects ψ_k but $\mathbf{d} \in A_{k-1}$ does not. Then by the Potential Cut Lemma (9e.4), statement (1), we have $\mathit{cap}(\psi_k(\mathbf{c})) \geq \mathit{flow}(\psi_k(\mathbf{d}))$. But $\mathit{cap}(\psi_k(\mathbf{c})) \leq \mathit{cap}(\psi_k(\mathbf{d}))$ by the choice of $\mathbf{c}$, so $\mathit{cap}(\psi_k(\mathbf{d})) \geq \mathit{flow}(\psi_k(\mathbf{d}))$, and $\mathbf{d}$ protects ψ_k after all. Thus line 7 of Algorithm C correctly implements line 3 of Algorithm A.

There are now two cases to consider. If the configuration $\mathbf{c}$ does protect ψ_k, then so do all configurations in A_{k-1}. Therefore Algorithm A sets A_k to A_{k-1}, and Algorithm C does not change I, so we have $C_{k-h} = A_k$ as desired. On the other hand, if $\mathbf{c}$ does not protect ψ_k, then Algorithm C adds to I the constraint derived from ψ in the configuration $\mathbf{c}$. This constraint is precisely Λ_k. □

We conclude that the configurations that obey the final constraint system I in Algorithm C are precisely those in A_m. (If the design system adds extra constraints to I, some configurations in A_m may be excluded.) Theorem 9e.1, which characterizes A_m, now implies that every configuration obeying I is routable, and that the constraints I are optimal, unless the constraints are allowed to define a disconnected region of configuration space. Finally, line 10 of Algorithm C finds an optimal configuration obeying the constraint system I. The resulting sketch is guaranteed to be routable, and hence Algorithm R can regenerate the layout. This completes the proof that the compaction algorithm is correct.

Optimizations of Algorithm C

Both the time and space performance of Algorithm C can be improved by reducing the size of the adjacency graph. One therefore wishes to choose gates in such a way as to minimize the number of crossings between traces and gates. Although we required the gates to form a partition of the sketch, one can get by with fewer. If the routing region is connected, a minimal set of gates is such that the set of points in the routing region but not on any gate is simply connected. Equivalently, if islands and gates are considered as the nodes and arcs, respectively, of a graph, then this graph should be a tree. One must be careful, however, to keep track of the direction of every crossing among the traces, gates, and terminals. The removable nodes and edges of the intersection graph depend upon these directions of crossing in a somewhat complicated manner. In essence, one must ensure that when modifying the intersection graph, the traces can be rerouted to reflect the new structure.

A minimum-cost spanning tree algorithm can be used to find a set of gates that cross as few traces as possible. Every horizontal cut between different islands is a potential gate, but we may restrict our attention to horizontal cuts that are incident on feature endpoints. There are at most $O(|F|)$ such cuts, and they can be thought of as the arcs of a graph H over the islands. The cost of a cut will be the number of crossings of the cut by traces in the original sketch; costs can be computed efficiently using a scanning algorithm as in Section 1D. The gates are chosen to be the arcs in a minimum-cost spanning tree of the graph H.

Another way to speed up Algorithm C is to ignore potential cuts that cannot generate constraints. For example, if a potential cut ϕ_{pq} has minimal capacity in the initial configuration, it cannot generate a constraint. This observation follows from statement (1) of the Potential Cut Lemma. More generally, if a potential cut is occluded in such a way that it cannot become a cut before reaching a minimum of capacity, then this potential cut may be ignored. Lemma 9e.4 (or more generally, Lemma 9e.3) can be applied in many other ways to justify the omission of potential cuts. For example, I showed in [29] that if the wiring norm is rectilinear—that is, if $\|(x,y)\| = \max\{|x|, |y|\}$—and the features are all horizontal or vertical, then the critical potential cuts may be omitted altogether.

None of these improvements affect the fact that Algorithm C requires $\Omega(|F|^3)$ time, not just in the worst case, but in almost every case. To reduce this amount, one must avoid considering most of the potential cuts. Most constraints in practice are likely to be local, so one can try to ignore all potential cuts of sufficiently large height. If one solves the constraint system before evaluating all the potential cuts, and the routing algorithm succeeds, then compaction may be terminated. If the routing algorithm fails, more potential cuts must be considered. A good heuristic for exploiting locality could reduce the average-case running time to quadratic or less, though the leading constant might be large.

Ultimately, the slowness of Algorithm C is due to its generality. The islands in a sketch compaction problem allows may be bound into modules in an arbitrary way, whereas in many cases of interest only local connections are needed. When all features are independent, as usually occurs in the compaction of routing channels, simpler and faster techniques are available that still insert all useful jogs automatically [59].

Wire length minimization

Usually when performing compaction one would like to improve wire lengths as well as layout area. Algorithm C minimizes trace lengths in a trivial sense, namely that the wires make no unnecessary detours. Because it uses Algorithm R, the lengths of traces are minimal given the positions of the features that it supplies. By default Algorithm C moves each module as close to the left-hand wall as possible, which will often be far from optimal. But it can be modified to support whatever wire-length minimization technique you favor. The constraint graph constructed by Algorithm C specifies the set of acceptable output configurations. Solving the constraint system with a longest-path algorithm determine the minimum separation between the walls. One can add the constraint that the walls be separated by that distance, thus defining a smaller set of acceptable configurations. One may then choose a configuration in this set by any desired means. If one can estimate the effects of configuration on total wire length, then one can find a configuration that nearly minimizes wire length. The problem of finding a good heuristic for making this estimate is open, but probably not too difficult.

Summary

I have presented a polynomial-time algorithm for one-dimensional layout compaction with automatic jog insertion. It works whenever layouts can be partitioned into layers such that wires on two different layers interact only via modules that are present on both. The algorithm takes its input as a set of proper sketches, one for each routing layer, and produces output in the same form. (For practical purposes, this means the input must be a legal layout.) Algorithm C treats the special case of one routing layer, which is no easier to compact than many connected layers. Jog insertion is achieved by treating wires not as objects to be moved, but only as indicators of layout topology. Using the sketch routability theorem, the algorithm converts the wires into constraints on module positions that ensure that the wires have sufficient room to be routed. Having determined a new placement for the modules, it then invokes a single-layer router (Algorithm R) to restore the wires with as many jogs as necessary. The compactor thereby inserts all jogs that help to reduce the width of the layout. It may use more jogs than necessary, however.

The version of Algorithm C presented here substantially generalizes the compaction algorithms in my earlier papers [28, 29]. Those algorithms worked only in a grid-based wiring model, while Algorithm C allows features and traces to be other than rectilinear, to have different widths, and to be governed by an arbitrary piecewise linear wiring norm. These extensions were made possible, of course, by the theory of single-layer wire routing established in Chapters 1 and 8. Further extensions or reformulations of this theory, as we will discuss in the next chapter, should lead to further generalizations of Algorithm C. Unlike the reasoning that underlies Algorithm R, the correctness proof of Algorithm C is nearly modular. Having the requirements of Algorithm A spelled out in Section 9E means that changes to Algorithm C can be easily justified.

Chapter 10

Extensions of the Theorems and Algorithms

This chapter is all about the sketch model: the rationale behind it, the extensions and modifications it supports, how the sketch algorithms can be adapted to handle these extensions, and how sketches may be used to represent circuit layers. For the most part the proposed changes to the sketch model are orthogonal, meaning that they can be adopted or ignored independently. Since many facts about the sketch model will be illustrated by reference to the design model, in this chapter I use the terms 'wire' ar d 'trace' interchangeably.

Chapter outline

The chapter is divided into four major sections. The first chiefly concerns the representation of standard devices as parts of sketches. It presents a view of separation constraints based on the *convolution* of geometric regions, and relates it to the use of wiring norms to define which sketches are proper. It suggests how the sketch model may be modified so that separation constraints can be defined independently for all pairs of sketch elements. Finally, it describes how to change the Algorithms T and R so that the terminals of each trace are permitted to approach one another. Both these extensions are helpful for representing integrated circuit layers.

Section 10B examines the aspects of the sketch model that govern the shapes of traces: the wiring norm, the allowed shapes of features, and the fact that traces are not constrained to a grid. It first shows that the sketch model does, in fact, subsume the grid-based wiring model. If all the features of a sketch lie in a grid of unit pitch, measured in a rectilinear wiring norm, and all the elements of the sketch have width 1, then the traces may be constrained to the grid without affecting routability. Moreover, one can add a simple postprocessing phase to Algorithm R to ensure that every trace is routed within the grid.

Section 10B then considers wiring rules at the opposite extreme: curvilinear rules in which the wiring norm is not piecewise linear. The theory of single-layer routing does not change substantially if the wiring norm is, say, the euclidean norm, and if circular arcs are allowed as features. Even without appealing to a more general theory of wire routing, we can show that the design routability theorem holds also for curvilinear wiring norms, provided that fringes remain polygonal. Hence the sketch routability theorem admits the same generalization. The trick we use involves approximating the curvilinear norm by a polygonal norm. It thereby allows us to apply Algorithm R to sketches with curvilinear wiring norms, although its performance declines and it cannot quite minimize wire lengths.

Section 10C steps farther out and considers some major extensions of the sketch model and the sketch problems. These include: allowing the terminals of a trace to merge or pass through one another during compaction; allowing terminals to be line segments or convex polygons as well as points; and allowing traces to have more than two terminals. My conclusion is that although the extensions seem to be possible, the sketch model is not well suited to them, particularly the addition of extended terminals and multiterminal nets. In Section 10D I propose a new model of wiring that incorporates extended terminals and multiterminal nets in an elegant way. I then discuss the prospects for adapting my theory of single-layer wire routing and its attendant algorithms to the new model.

Development of the sketch model

Before describing various generalizations of sketches, I should explain some of the reasons why the sketch model has the properties it does. My advisor Prof. Leiserson and I originated the sketch model as a generalization of the grid-based wiring model used Leiserson and Pinter [22] and many others. We wished to consider wiring rules more general than grid models, and so we quickly abandoned the common convention that terminals are points on the boundaries of modules. Instead we decided to separate terminals from the modules they helped interconnect. The reason was to avoid introducing spurious cuts that might falsely indicate unroutability; see Figure 10-1 for an example. The desire for a clean routability theorem was the major motivation for most of my decisions concerning the sketch model.

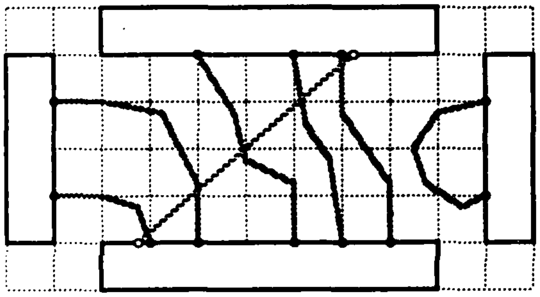


Figure 10-1. *Terminals are not points on other features.* If they were, some cuts in a routable sketch could be both nonempty and unsafe. Here the traces have width equal to the distance between adjacent dotted lines, and the unit polygon is square. The striped cut has a congestion of 5 but a capacity of only 4.5. Yet the traces can be routed.

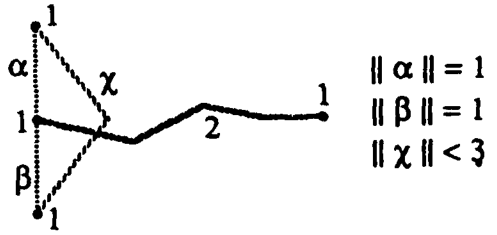


Figure 10-2. *Wires cannot be wider than their terminals.* **This example shows why: the sketch is unroutable, as the bent cut χ is unsafe and nonempty. All the straight cuts, however, including α and β, are safe.**

Another peculiarity of sketches is their requirement that each trace be no wider than its terminals. Others have made such an assumption to simplify design-rule checking [37]. My reason comes from the the sketch routability theorem. If the requirement were removed, this theorem would be false. Figure 10-2 shows the counterexample. The breakdown in the proof can be traced to Lemma 4f.3, which shows that the capacity of a major cut is no less than the capacity of its elastic chain. This lemma is used to prove Corollary 4f.5: that a safe sketch, whose major straight cuts are safe, has no unsafe, major, bent cuts. In Figure 10-2 this claim fails: the bent cut is unsafe, but the links of its elastic chain are safe. The reason is that in going from the bent cut to its elastic chain, the flow has decreased by the width of the wire, but the capacity has only decreased by the width of the its terminal, which in this example is smaller.

Self-avoidance

Perhaps the most puzzling aspect of the sketch and design models is the requirement that wires be self-avoiding. I am frequently asked why this condition exists, and whether for practical purposes it could be ignored. Unfortunately there are good reasons for being concerned with self-avoidance. The question is nevertheless a good one, because it seems that whenever a wire fails to be self-avoiding, a simple change to the topology would remove the offending loop of wire, improving the routing and avoiding design-rule violations. This hope is dashed by examples like Figure 10-3, which shows that two parts of a wire can approach too closely in a gap that is too narrow for the wire to be routed through. Programmers of computer-aided design tools have assured me that such situations should not be assumed to be absent in practical designs.

The first reason to require self-avoidance of wires is that the sketch and routing theorems depend upon it. Without it, the sketch of Figure 10-3 would be proper without being safe. A possible escape from this dilemma is to redefine the flow across a cut so that consecutive necessary crossings of a cut by the same wire contribute only one wire's thickness to the congestion. In order for this approach to work, one would have to allow wires to intersect themselves. Such a change would almost certainly cause more problems than it solved.

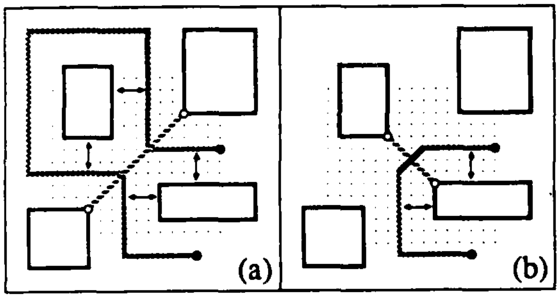


Figure 10-3. *Rerouting cannot ensure self-avoidance.* All elements have width 1, the unit polygon is square, and the distance between adjacent dots is 1/3. The sketch in (a) is improper because its trace fails to self-avoid; the striped cut has congestion 2 but capacity 5/3. Changing the topology to (b) does not help; the striped cut has congestion 1 but capacity 1/3.

In any case, there are eminently practical reasons to insist that wires be self-avoiding. One can make a case for an even stronger condition. Let us call a wire **strongly self-avoiding** if the union of its territory with those of its terminals does not separate the plane into two or more components. (Ordinary self-avoidance requires only that this union not separate one island of the sketch from another.) An equivalent definition is that a wire is strongly self-avoiding if its extent is simply connected. If a wire is not strongly self-avoiding, then two parts of the wire violate the design rule concerning wire-to-wire separation. This raises the possibility that a short between these parts would form a loop in the layout, which (I am told) could function as an unwanted antenna; or that while the wires are being laid down, the thin piece of resist between the nearby wire parts could break off and foul the circuit. For these reasons, some self-avoidance condition must be imposed on wires, and possibly on routing obstacles as well.

The property on which my definition of self-avoidance is based, namely divisiveness of design articles, has the benefit that it can be tested by looking at nondegenerate straight cuts. (See Lemmas 5e.4 and 6a.2.) In contrast, there is no natural set of cuts whose safety determines whether the extent of an article separates the plane. Another benefit of my definition is that the self-avoidance of ideal embeddings of wires and ideal realizations of traces is relatively easy to verify. Ideal embeddings and ideal realizations are strongly self-avoiding, but the proof is fairly difficult.

10A. Representation Issues

As it stands, the sketch model is not very close to the models that circuit designers actually use. Although grid models, which the sketch model subsumes, are acceptable for channel routing problems, they are poorly suited for representing transistors, the primary components of integrated circuits other than wires. Problems arise when trying to map the geometric design rules onto the sketch model. Usually the rules are separation constraints and overlap constraints among regions on various circuit layers. Some of the regions have no natural counterparts in the sketch model, and some regions must be represented as a set of sketch elements if that region needs to connect to wires. Nevertheless, with suitable extensions to the sketch model described here and in later sections, one can obtain approximations of real design rules. Though sketches cannot adequately represent most transistor structures, they can probably handle the interconnection of larger modules.

Convolution of regions

The rules governing proper sketches and designs are stated in terms of a global wiring norm. This approach has the virtue of simplicity, but it grew out of a more basic and more flexible view of geometric design rules, which I now describe. It begins with the assumption that wires, at least, are to be represented as paths, rather than regions of positive area, in order to define homotopy relations among wires. Hence we must relate the abstract wire, which we think of as a path or its image, to the region that the wire is to occupy in the circuit. We must also convert the design rules among these regions to design rules among the abstract wires.

A natural approach is to define the regions that wires occupy, and the regions that they are forbidden to occupy, using the operation of *convolution*. For the purposes of this section, a **region** is a subset of the plane R^2. The **convolution** of two regions A and B, which I denote $A + B$, is the set of all vector sums of points in A with points in B, namely,

$$A + B = \{ a + b : a \in A,\ b \in B \}.$$

We consider wires whose shape can be described as the convolution of a **centerline**, call it C, with a region W that contains the origin 0, as shown in figure 10a-1. The region $C + W$ occupied by the wire can be obtained by sweeping the **brush** W along the centerline C, keeping the origin of W on C. The required separation between wires may also be described using convolution. If R_1 and R_2 are the regions occupied by two wires, there may be a region S_{12} such that R_1 and R_2 are sufficiently separated if and only if $R_1 + S_{12}$ does not intersect R_2. This kind of design rule is quite general. Each wire can have a different brush, and each pair of wires can have a different region defining their required separation. Self-avoidance can also be described using convolution; there may be regions S_{11} and S_{22} such that $R_1 + S_{11}$ and $R_2 + S_{22}$ are not allowed to divide the plane.

The convolution conditions become somewhat simpler if we assume that the regions W_i and S_{ij} have inversion symmetry. If B is any region, we define the set $-B$ to be $\{ -b : b \in B \}$. Suppose $W_i = -W_i$ and $S_{ij} = -S_{ij}$ for all i and j. Then the conditions

$$((C_1 + W_1) + S_{12}) \cap (C_2 + W_2) = \emptyset \qquad \text{and} \qquad (C_1 + W_1) \cap ((C_2 + W_2) + S_{12}) = \emptyset$$

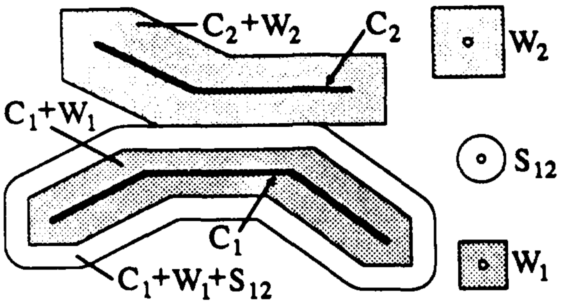


Figure 10a-1. *Examples of convolution.* The small circle inside each of the regions W_1, W_2, and S_{12} marks the location of the origin. We take these regions to be open so that the convolved regions are open.

are equivalent, and are also equivalent to the condition

$$(C_1 + T_{12}) \cap C_2 = \emptyset \quad \text{where} \quad T_{12} = W_1 + S_{12} + W_2.$$

(Note that convolution is associative and commutative.) So the brushes W_i and separation regions S_{ij} may be discarded in favor of the regions T_{ij}.

One can take the convolution idea a step further and consider the constraints that arise among the wires crossing a cut in a sketch. Let $\overline{pq}$ be a cut between two obstacles P and Q, which for simplicity we consider to be points. As we saw in Chapter 8, there is a definite sequence of traces that must cross $\overline{pq}$; they have a definite ordering from P to Q. Let $C_1, \ldots, C_n$ denote the centerlines (i.e., images) of these traces, and put $C_0 = P$ and $C_{n+1} = Q$. For $1 \leq i \leq n$, let T_i be the region defining the required separation between C_i and C_{i+1}. If the ith and $(i+1)$st traces are the same, then T_i determines the self-avoidance requirement for that trace. The centerline of the first trace must satisfy $(P + T_0) \cap C_1 = \emptyset$. Assuming that the sets T_i are well behaved (they should be simply connected and should contain the origin), the closest C_1 can come to P is the edge of the region $P + T_0$. Similarly, even if C_1 wraps tightly around $P + T_0$, the closest the second centerline C_2 can come to P is the edge of $P + T_0 + T_1$. (This conclusion is accurate only if P does not interact with C_2 through C_1.) Thus the ith centerline C_i is forbidden to enter the region $P + T_0 + \cdots + T_{i-1}$. This region is a *barrier* for C_i in the sense of [49]. The cut $\overline{pq}$ is *safe* if and only if the following condition holds:

$$(P + T_0 + T_1 + \cdots + T_n) \cap Q = \emptyset. \tag{10-1}$$

One could probably build a theory of single-layer wire routing on this basis. I have chosen a simpler foundation to avoid making the proofs of the routability and routing theorems any more difficult than they already are.

Relation to wiring norms

Under certain common conditions the design rules defined via convolution can also be derived from a wiring norm. The basic requirement is that all the regions

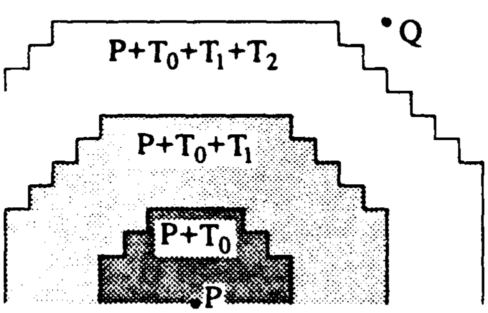


Figure 10a-2. *Barriers as convolved regions.* Here there are three necessary crossings of the cut from P to Q, and for each crossing there is a barrier around P. The minimum separation between wires is determined by convolving with a nonconvex polygonal region $T_0 = T_1 = T_2$. This approach can be used to model the routing of unit-width wires in a quarter-integer grid [49].

T_{ij} defining the separations between centerlines be multiples of a convex, open, symmetric region T. If T is a region and $r \geq 0$, let rT denote the dilation of T by the factor r: the region $\{ r \cdot x : x \in T \}$. If T is convex, open, nonempty, and $T = -T$, then we can define a norm $\| \cdot \|_T$ by

$$\|x\|_T = \inf\{ r \geq 0 : x \in rT \}.$$

This norm has the property that rT is the region $\{ x \in R^2 : \|x\|_T < r \}$. As a consequence, the convolution $rT + sT$ equals $(r + s)T$. If we suppose that each region T_i has the form r_iT, then the condition $(C_1 + T_1) \cap C_2 = \emptyset$ is equivalent to the condition $\|C_1 - C_2\|_T \geq r_1$. (The quantity $\|C_1 - C_2\|_T$ equals the infimum of $\|x\|_T$ over all x in the convolution $C_1 - C_2$.) Condition (10–1) above is equivalent to $\|P - Q\|_T \geq \sum_{i=0}^{n} r_i$. If every centerline C_i of a trace or obstacle can be assigned a width w_i such that the required separation T_{ij} between C_i and C_j is $\frac{1}{2}(w_i + w_j)T$, then we are back to the sketch and design models, with $\| \cdot \|_T$ as our wiring norm. The width of an element accounts not just for the size of its brush, but also for its required separation from other elements.

Only rarely will the use of a fixed wiring norm be too restrictive. Ideally the design rules would all be isotropic, and the design system would take full advantage of them by permitting circular arcs in centerlines and drawing components with circular brushes. In this case the wiring norm would be the euclidean norm. (Curvilinear wiring norms will be discussed in the next section.) But usually, for simplicity of programming and compatibility with manufacturing equipment, the design system deals only with polygonal regions, or only with rectangles. In this case the brushes and separation regions will all be multiples of a standard polygonal region, typically a square or an octagon, and the polygon bounding this region will be the unit polygon of the wiring norm.

But there is a problem in stipulating that the minimum separation between two components be purely a function of their widths. Consider a typical MOS technology that represents transistor gates by overlapping regions of diffusion and polysilicon. Although diffusion and polysilicon are usually thought of as different layers on the chip, for the purposes of routing they must be combined, since wires of the two materials must not cross except where a transistor is to be placed. The minimum separation requirements between two polysilicon regions, between polysilicon and diffusion, and between two diffusion regions may all differ. Any design rule that was blind to differences between materials would have ιo be very conservative.

Components of differing materials

To treat the possibility that different components are made of different materials, the sketch model must be generalized. Instead of assigning each element a fixed width, a sketch will include a matrix of minimum distance constraints among the elements of the sketch. Let us denote the required separation between elements i and j by $s(i,j)$. We assume that for all i, j, and k we have $s(i,j) > 0$ (positivity), $s(i,j) = s(j,i)$ (symmetry), and $s(i,k) \leq s(i,j) + s(j,k)$ (the triangle inequality). No longer will each trace and island have a fixed territory. Instead two distinct elements i and j of a sketch will be considered properly separated if the distance between them (in the wiring norm) is at least $s(i,j)$, or if one is a terminal of the other. Similarly, the trace i will be considered self-avoiding if the set of points lying $\frac{1}{2}s(i,i)$ units or more from that trace has only one component that contains features. As before, a sketch is proper if its elements are properly separated and its traces are self-avoiding. We used to insist that the terminals of a trace have the same width, and that this width equal or exceed the width of the trace. This demand translates into the following: for each trace k with terminals i and j, we have for each element l the relations $s(i,l) = s(j,l) \geq s(k,l)$.

Some definitions concerning cuts must change also. The capacity of a cut will no longer account for the widths of its endpoints, since the contributions of those endpoints are uncertain. Instead we put the capacity of a cut equal to its arc length in the wiring norm. The congestion of a cut will now depend upon the sequence of traces that necessarily cross the cut. Let the endpoints of the cut lie on elements number e_0 and e_{n+1}, and suppose that the content of the cut is $\langle e_1, \ldots, e_n \rangle$. Then the congestion of the cut is defined to be $\sum_{i=0}^{n} s(e_i, e_{i+1})$. The cut is considered empty if $n = 0$ and $e_0 = e_1$, and safe if its congestion does not exceed its capacity.

I conjecture that if these changes are made to the sketch model, then the sketch routability and routing theorems continue to hold. There is strong support for this claim, I believe, from an observation concerning the proofs of the design routability and routing theorems. The key results concerning flow (Proposition 4d.2, Proposition 4f.1, Lemmas 5c.2 and 5d.2, and Proposition 6a.3) can all be reformulated in terms of content rather than flow. In other words, they never rely on the flow across a cut (in the usual design model) being independent of the ordering of the necessary crossings of that cut. To prove the conjecture, one would probably have to extend the design model by making changes corresponding to those I have suggested for the sketch model, and repeat the development of Chapters 4 through 8.

(Some aspects of the design model that have no counterparts in the sketch model, such as the definition of the flow across a half-cut, would also need to change.) I have not attempted to carry out this program, but I have little doubt that it would eventually be successful.

If the design routing and routability theorems remain correct, then Algorithms T and R can be generalized to the new model. The required changes are simple and do not affect the worst-case performance bounds. In essence, one replaces the summing of element widths by the summing of element-to-element spacings. This replacement occurs in four places: in the construction of doorways by Algorithm R; in the determination of cable widths in the condensed RBE; in the data structure of Algorithm T that contains trace segments (which we called WS); and in the main loop of Algorithm F. Each cable in the condensed RBE must store, in addition to its width, the identities of the strands at the left and right edges of that cable, so that Algorithm T may compute the proper spacing between the strands of this cable and those of another. Processing this information still takes only constant time per cable. Likewise, the preprocessing for Algorithm F, which normally stores the lengths of the shortest paths between various nodes in the adjacency graph, must also keep track of the first and last traces along those shortest paths.

Nonlocal constraints

A further extension would remove the assumption that the minimum separations $s(i,j)$ satisfy the triangle inequality. Such an extension may be necessary for handling complementary MOS technologies, in which there are large separation constraints between n-type and p-type transistors, far larger than the typical spacing between wires. Such nonlocal interactions probably cannot be handled at all if they involve traces. But if they involve only fixed devices, represented by islands, then one can divorce the question of whether devices are properly separated from the issue of routability. One would simply precede one's routability test by a straightforward test, taking perhaps $O(n^2)$ time, to ensure that each pair of features satisfies its minimum separation requirement. Routing would not be affected.

Representing devices in a sketch

A method of handling traces and features of differing materials would remove the major hindrance to accurate representation of integrated circuit layers by sketches. (In contrast, printed circuit board layers are much simpler, and the sketch model as given in Section 1A is probably adequate to describe them.) In what follows I assume that such a method is available.

By far the most common devices in an integrated circuit are contact cuts and transistors. On any particular layer, a contact cut is nothing more than a convex region to which a wire may connect. This region is typically circular or square, depending on the wiring rules in effect, and in general it may be given the shape of the unit polygon (or circle) of the wiring norm. Thus it can be represented as a pointlike feature. Typically its width is greater than that of the attached wire; this is permitted by the sketch model. Some contacts, like the "buried" contacts in MOS technologies, connect two wires on the same routing layer. Like transistors, these must be represented as multiterminal devices.

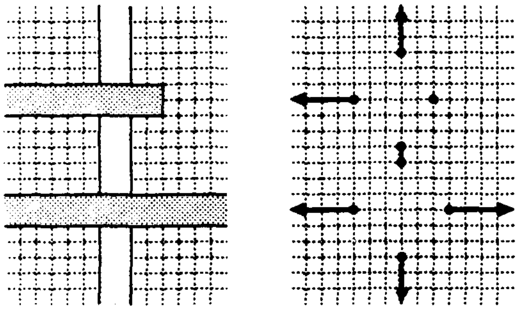


Figure 10a-3. *Representing transistors.* At left, two polysilicon wires (dark shading) cross a diffusion wire (light shading) to form typical enchancement mode transistors. The situation can be represented in a sketch using four features per transistor: two of type diffusion and two of type polysilicon. This example uses Mead/Conway design rules [31] with λ equal to the grid spacing.

Transistors are built out of sets of features. A transistor is usually a three-terminal or four-terminal device, and so its representation must include at least three or four pointlike features to which traces can connect. In many systems the gate of a transistor can be determined implicitly by the crossover between polysilicon and diffusion wires, but since sketches prohibit crossings between wires on the same layer, such an approach is ruled out. Hence the sketch may need additional features to occupy the active area of the device, and prevent any other traces and features from approaching too closely. Finally, the transistor structure must be internally consistent; its islands must be properly separated. Figure 10a-3 shows how the simplest kind of transistor might be represented.

Unfortunately, transistor structures involving butting contacts, implantation regions, and so on are much harder to represent in sketches. To avoid design rule errors one is forced into a very conservative representation. Moreover, because terminals are points, the transistor structures are relatively inflexible. During compaction one probably cannot allow movement of the connection points relative to one another, even when such movement might be desirable; all the islands forming the transistor should be placed in the same module. Some flexibility can be regained using extended terminals, however; see Section 10C.

Terminal merging

Some representations of devices work well in the presence of wiring alone, but less well in combination with one another. The reason is that two devices may sometimes approach more closely than their representations would indicate. For example, the two transistors in the left-hand panel of Figure 10a-3(a) are farther apart than necessary, but their terminals in the right-hand panel are too close. Figure 4 in the Introduction contains many other examples of modules overlapping. One would like to permit such overlaps, since they actually violate no design rules, but the sketch model prohibits it. The culprit is the requirement that the terminals of a trace have disjoint territories. If we remove this restriction, and allow the territories of each trace's terminals to merge, then we can route circuits like those depicted in the Introduction, without invoking special-purpose representations for groups of devices. I call this process **terminal merging**, because if we take the idea to its natural conclusion, it allows the terminals of a trace to coalesce during compaction.

We already have most of the machinery needed for terminal merging. By default the design model allows the terminals of a wire to be arbitrarily close, provided that the wire remains self-avoiding. In fact, when we began translating the results of the design model, in Chapter 7, we had to make special allowance for this difference between sketches and designs. We could just as well change the sketch model to permit terminal merging. This change would have only one major drawback: a complication of Algorithm C, the sketch compaction algorithm, and its proof of correctness. We discuss this issue further in Section 10C.

For now let us consider how terminal merging would affect routing and routability testing. As noted in Section 6C, whether terminals are permitted to approach one another has no affect on the ideal embedding of a design, and consequently it does not affect the ideal realizations of a sketch either. Thus Algorithm R is indifferent to terminal merging. Algorithm T, on the other hand, would have to avoid checking the degenerate cuts—those which correspond to degenerate cuts in a design. As it turns out, the only straight, degenerate cuts that are not also empty are straight cuts that coincide with rubber bands. Hence Algorithm T can be easily modified to permit terminal merging.

10B. Wiring Rules and Wiring Norms

In this section we consider four different wiring rules that may be attached to the sketch model. One asks that wires be composed of horizontal and vertical segments only. The next is even more restrictive: it requires that the wires run in a grid. In another the wiring norm is euclidean, or any other easily computable norm that is not piecewise linear, and wires are allowed to contain curves as well as line segments. The fourth is the same, but it also allows features to contain curved pieces. In each case we examine the effects on the sketch routing and routability theorems, and on the performance of Algorithms T, R, and C. As usual, I shall characterize wiring norms by the locus of points of norm 1, the *unit polygon* or *unit circle* of the norm. Thus a piecewise linear norm is "polygonal", other norms are "curvilinear", and the norm attached to the grid model is "rectilinear".

Restrictions on wire segments

Suppose we require that the traces in a proper sketch consist only of horizontal and vertical segments. This requirement is necessary if the fabrication process or the design system can handle only rectangles whose sides are aligned with the coordinate axes. Let us assume, therefore, that all the line segments representing the features in our sketches are horizontal and vertical also. The appropriate wiring norm is rectilinear: define $\|(x,y)\|$ to be $\max\{|x|,|y|\}$.

Under these assumptions the sketch routability theorem still holds, and the basic sketch algorithms continue to work. Clearly it remains true that every unsafe sketch is unroutable. For the converse, I present a method for transforming the ideal realizations of the traces in a safe sketch, which may contain diagonal segments, into realizations consisting of horizontal and vertical segments only. This rerouting method appeared in an earlier paper with Leiserson on sketch routing [21], and can be added as a postprocessing phase to Algorithm R with a loss in performance of at most a constan* factor. Consequently the other sketch algorithms need not be changed. Aside from using Algorithm R as a subroutine, Algorithm C relies only on the sketch routability theorem and basic properties of the sketch model. Algorithm T is likewise unaffected.

The sketch routing theorem, on the other hand, weakens s· mewhat. In general there are many feasible realizations for a trace that consist of horizontal and vertical segments and have minimal length under those conditions. In short, minimum-length feasible realizations are no longer unique. But the traces in a safe sketch still have minimum-length realizations that form a proper sketch. The modified Algorithm R computes such minimum-length realizations.

The rerouting process

Given the ideal realization of a sketch, the usual output of Algorithm R, we reroute each trace downward onto its struts as shown in Figure 10b-1. Only one trace θ need be considered at a time. We first identify the joints of θ that are **stationary**. Recall from Section 1D that an ideal trace is supported at each joint by a strut, which is part of a diagonal cut. With a square wiring norm the diagonal slopes are ± 1, and hence each strut points either upward or downward and either leftward or rightward. A joint is stationary if either the strut supporting θ there is upward, or a segment ending at that joint is horizontal or vertical. The stationary joints divide θ into **flexible** subpaths, each of which is either a single segment or a chain of segments whose angles lie in the same quadrant: either all the segments in the chain point upward and leftward, or they all point upward and rightward, et cetera. (These facts follow from the results of Section 7D, because ideal traces are tracks.)

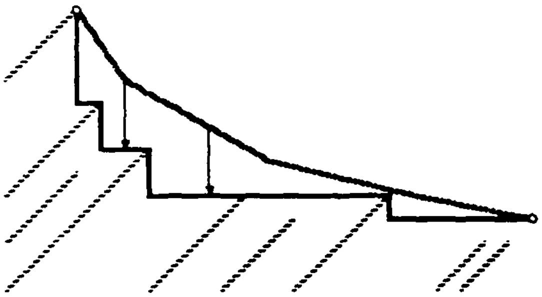


Figure 10b-1. *Routing a flexible subpath onto its struts.* The flexible subpath (grey) is replaced by a rectilinear path (black). Only the struts (striped lines) of a particular diagonal slope are considered: that with opposite sign to the sign of the slopes of the segments in the subpath.

Each flexible subpath of θ is rerouted downward onto certain of its upward struts, keeping its endpoints fixed. A flexible subpath that contains only a single horizontal or vertical segment may be left alone. Consider now a flexible subpath whose segments all have positive slope. This subpath is to be rerouted onto struts that point upward and leftward. Which struts are they? The endpoints of the subpath, those that are not terminals, are supported by struts of slope -1. Hence this subpath passes through a particular portion of the corridor for θ corresponding to the diagonal slope -1. Consequently there is a well-defined sequence of struts of slope -1 that constrain this flexible subpath from below. If this flexible subpath consisted instead of segments of negative slope, we would reroute it onto struts that point upward and rightward in the same way.

Though I have no intention of providing any formal proofs in this chapter, it should not be difficult to believe that this method works, and works efficiently. First of all, there is no chance of changing the routing topology by moving a flexible subpath across a feature, because any such feature would give rise to a strut constraining that subpath. For the same reason, the new routings are actually traces—intersecting no features except their terminals—and they also remain properly separated from all features other than their terminals. Second, the rerouting causes no two flexible subpaths to collide. For suppose it did; suppose it pushed an upper subpath downwards onto or through a lower subpath, as shown in Figure 10b-2. By symmetry we may assume that the upper subpath originally consisted of segments of negative slope. Then the lower subpath would have an upward, rightward strut that intersects the other subpath as well. But the feature that gives rise to this strut also gives rise to a longer strut for the upper subpath. Hence the upper subpath could not be rerouted down onto the lower one. (This argument is reminiscent of the proofs of Lemmas 5c.2 and 5c.3, and one could formalize it by adapting those proof techniques.) Finally, the rerouting is efficient; it requires only constant time per strut, and hence consumes no more time and space, up to constant factors, than Algorithm R normally does.

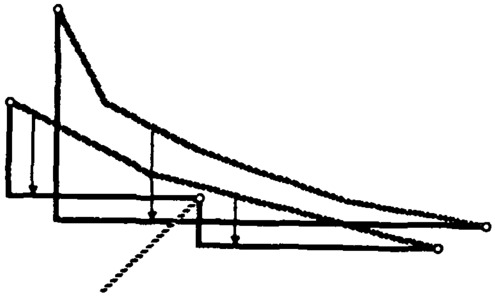


Figure 10b-2. *No flexible subpaths collide.* If one were somehow rerouted down through another, it would cross a strut for the lower one (striped path) that is part of a strut for the upper one.

One last claim is that the traces output by the rerouting process are as short as possible. This claim is harder to verify, because it depends on the unproven fact that no feasible realization of a trace is shorter than its ideal realization in *any* norm. Suppose we measure wire length with the taxicab or L^1 norm. The ideal realization of a trace has minimum length in this norm among all feasible realizations of that trace. Since the rerouting process does not affect arc length in the taxicab norm, the rerouted traces are still optimal in this sense.

The grid model

From the model in which a proper sketch consists only of horizontal and vertical segments, the grid model is but a short step away. Let R denote the real line and Z the set of integers. The **grid** relevant to single-layer routing is the set $\{ (x,y) \in R^2 : x \in Z \text{ or } y \in Z \}$ of points with at least one integer coordinate. The lines it contains are called **gridlines**, and the points $Z \times Z$ where they intersect are called **gridpoints**. The grid model for sketches makes the following assumptions: all features in a sketch lie in the grid; all feature endpoints are gridpoints; the width of each element in a sketch is an odd integer; and the wiring norm is the rectilinear one. It mandates that a sketch is not proper unless each of its traces lies within the grid.

Despite the additional restrictions imposed by the grid model, it need not be treated any differently. The reason is that the rerouting stage of Algorithm R still produces proper realizations. Because the width of every sketch element is an odd integer, all the struts have integral length in the wiring norm, and so their endpoints are gridpoints. Consequently the rerouted traces all lie in the grid. The sketch routability theorem continues to hold, the sketch routing theorem holds in its weak form (trace lengths can be simultaneously minimized, but not uniquely), and Algorithm T is unchanged.

Algorithm C continues to work because it never considers a nonintegral displacement for any module. The additional restrictions on sketches could only cause Algorithm C to err by producing an improper sketch as output. We show that this event never happens. The congestions and capacities of all cuts, as well as the coordinates of every feature endpoint, are integers. Hence in each constraint that Algorithm C adds to its constraint system, the constant is an integer. Therefore all paths through the constraint graph have integral length, and so the configurations that Algorithm C computes involve integer dispacements only. Since the input sketch is assumed to be routable, its features must lie in gridlines and their endpoints on gridpoints. The same is true of the sketch that Algorithm C gives to Algorithm R, and so the output of Algorithm C is proper.

Curvilinear models

Having gone nearly as far as possible in restricting the traces in a proper sketch, from now on we consider allowing them some liberties that are lacking in the original sketch model. The simplest of these is the ability to contain arcs as well as segments. Such an ability is useless as long as features are line segments and the wiring norm is polygonal, so we will consider relaxing both of these assumptions. First we assume that the wiring norm is not polygonal.

Although we have no mechanism for dealing with curvilinear traces, there is a trick that converts a curvilinear norm to a polygonal norm for the purpose of routing. Using this trick we can prove that the sketch routability theorem holds for any wiring norm. For the sake of simplicity we take the wiring norm to be the euclidean norm, as there is probably no call for any other nonpolygonal norm.

Approximating the wiring norm

We construct the surrogate norm from the features in the sketch to be routed. Let S^1 be the unit circle $\{ x \in R^2 : |x| = 1 \}$, and let Λ be a set that contains, for each feature endpoint p and each feature Q having a cut to p, the line segment $\overline{pq}$ from p to Q that minimizes $|q - p|$. Let C be a convex polygon, symmetric about the origin, which does not intersect the inside of S^1. Suppose further that for each line L through the origin that is parallel to a line segment in Λ, two sides of C are tangent to S^1 at the intersections of S^1 with L. Such a polygon is easily created, as shown in Figure 10b-3. Start with any symmetric, convex polygon whose inside contains that of S^1. Then for each segment λ in Λ, intersect that polygon with the two lines perpendicular to λ and tangent to S^1, yielding either one or three polygons; take the one that encloses the origin. The polygon C that results from this process is the unit polygon of a norm $\|\cdot\|$ defined as follows: for any point $x \in R^2$, the quantity $\|x\|$ is the number $r \geq 0$ such that $x = rc$ form some $c \in C$. You may check that $\|\cdot\|$ is in fact a norm. It is stronger, or more restrictive, than the euclidean norm, in that $\|x\| \leq |x|$ for all points $x \in R^2$.

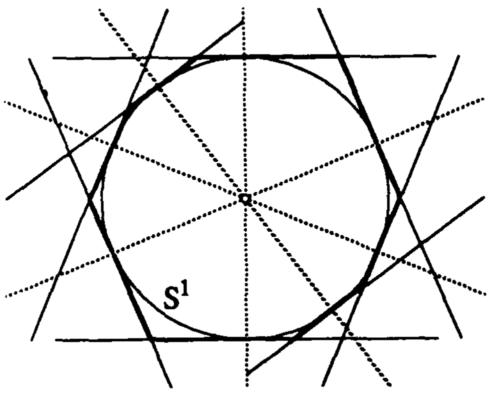


Figure 10b-3. *Constructing the unit polygon.* The striped lines take the slopes of all the critical cuts and all the line segments that would be critical cuts if their middles crossed no features. We circumscribe about the unit circle S^1 a polygon (dark lines) that is tangent to S^1 wherever these lines intersect S^1.

The new polygonal norm has two key properties that allow it to substitute for the wiring norm. The critical cuts of the sketch are the same in both norms, and they have the same length in both norms. In each norm the critical cuts are those cuts that begin at a feature endpoint p and travel to the closest point on another feature Q, with ties broken using the euclidean norm. Hence it suffices to show that if the point $q \in Q$ minimizes $|q-p|$, then it also minimizes $\|q-p\|$, and that the two distances $\|q-p\|$ and $|q-p|$ are equal. Let $\lambda \in \Lambda$ be the line segment $\overline{pq}$. Let O be the circle $\{x : |x-p| = |q-p|\}$ centered at p and passing through q, and let P be the polygon $\{x : \|x-p\| = |q-p|\}$. By the construction of C, the polygon P is tangent to O at q. Thus $\|q-p\| = |q-p|$. If T is the line through q perpendicular to λ, then by the choice of q, the feature Q does not intersect the side of T that contains p. Also T is tangent to O at q, and hence is also tangent to P at q. Since P is convex, the only side of T it intersects is that containing p. Hence no point of Q is closer to p than q in the norm $\|\cdot\|$.

Now we can see why a sketch whose critical cuts are safe is still routable. Switching from the euclidean norm $|\cdot|$ to the polygonal norm $\|\cdot\|$, the critical cuts do not change, and neither do their lengths. Hence their capacities remain unchanged, as do their flows, emptiness, and safety. Thus the sketch whose critical cuts are safe in the norm $|\cdot|$ has the same property with respect to the norm $\|\cdot\|$, and hence is routable in the norm $\|\cdot\|$, by the sketch routability theorem. In other words, our sketch has a realization that is proper with respect to the new wiring norm. The unit polygon of the norm $\|\cdot\|$ circumscribes the unit circle of the norm $|\cdot|$, and hence every element of our sketch has larger extent in the former norm than in the latter. (The polygonal norm is stronger.) Therefore, as in Lemma 8b.1, the realization that is proper in the polygonal norm is also proper in the euclidean norm.

A similar argument proves the other direction of the sketch routability theorem for the euclidean wiring norm. Given a sketch containing an unsafe, nonempty, straight cut λ, critical or not, one can construct a *weaker* polygonal norm in which λ remains unsafe. Let L be the line through the origin parallel to λ, and let P and Q be the islands containing the endpoints of λ. It suffices to inscribe a convex, symmetric polygon C in S^1 whose intersections $\pm x$ with L satisfy

$$\mathit{flow}(\lambda) > \frac{|\lambda|}{|x|} - \mathit{width}(P)/2 - \mathit{width}(Q)/2.$$

This condition ensures that in the norm whose unit polygon is C, the cut λ remains unsafe. Hence by Proposition 8b.3, our sketch is unroutable in the new polygonal norm. Since this norm is weaker than the euclidean norm, meaning that it gives rise to smaller extents for sketch elements, any sketch that is proper in the euclidean norm is also proper in the polygonal norm. Consequently our sketch is unroutable in the euclidean norm also.

Because the sketch routability theorem carries over to curvilinear wiring norms, so do Algorithms T and C. The sketch routing algorithm, however, fares less well. The trick of replacing the wiring norm by a polygonal norm is computationally effective, but the complexity of the resulting norm slows down the routing process greatly. The time and space complexity of Algorithm R are both proportional to the number of diagonal slopes, which can be up to $\Theta(n^2)$ if most features are visible from most other features. Hence Algorithm R could use up to $O(n^4 \log n)$ time and $O(n^4)$ space trying to route with the euclidean norm, and the result would not even minimize wire length.

Other approaches to curvilinear wires

A better approach to the routing problem is needed if routing with a euclidean metric is to be practical. Storb et al. [33] have recently developed a routing algorithm for sketches in the euclidean metric that combines the ideas of scanning over the rubber-band equivalent with the barrier construction methods of [52]. In a sense, their algorithm routes wires in the simply connected covering space of the routing region. It runs in time $O(|F|^2\,|T|)$ on a sketch (F, T). Another approach is to use relaxation starting from the rubber-band equivalent, inflating wires to their full width one at a time, moving other wires out of the way, and keeping all the wires tight. This approach seems to work well in practice [36], especially if performed incrementally as the sketch is being input. The worst-case running time of this method is as yet unknown.

We conclude that the problem of finding a nearly optimal algorithm for sketch routing in a curvilinear wiring norm is still open. (Perhaps the best idea is to drop the euclidean norm in favor of some reasonable, prespecified polygonal approximation.) My *theorems* of single-layer routing probably do extend to arbitrary wiring norms and quite general feature shapes, however, as I now discuss.

Arcs in traces and features

Any statements I make concerning single-layer routing with nonpolygonal wires and features must be somewhat speculative. The only way to justify them on the basis of present knowledge about sketches would involve a limiting argument like that relating sketches to designs. As we know, such arguments are extremely tedious. Another approach, which is perhaps no shorter but requires no new ideas, is to strengthen the existing theory, replacing 'piecewise linear' with 'piecewise smooth' or some intermediate condition. In my view, nothing but a heap of technical detail stands in the way of this improvement, but those readers who are less than intimately familiar with Chapters 2 through 9 may not share my confidence. So the generalizations that I am about to propose must be taken as conjectures.

The design routability and routing theorems hold under any wiring norm, provided that wires are permitted to include canonical paths in sets of the form $\{\, x : \|x - F\| = c \,\}$, where F is a fringe and $c > 0$ is a constant. The same is true even if fringes are the images of arbitrary piecewise smooth loops. (As before, the inside of each terminal must be a convex set.) This claim is really more general and less obvious than necessary. One important special case may be easier to swallow: the design routability and routing theorems are true in the euclidean wiring norm if wires and fringes can contain circular arcs of arbitrary radii. These conjectures have counterparts in the sketch model as well. In particular, I claim that the sketch routability and routing theorems are true in the euclidean wiring norm if features and trace segments can be circular arcs.

The sketch algorithms fare more poorly than the theorems when the sketch model is generalized. As we know, the sketch routing algorithm is unable to operate with curved elements. Algorithm R is grounded in the idea of building ideal realizations out of partial realizations. When the wiring norm is not polygonal, this idea cannot be applied, except by approximating the wiring norm.

At least while features remain straight, critical cuts can still be identified. Hence Algorithm T continues to work, as does Algorithm C, at least until the point where the output sketch needs to be routed. When features are not straight, however, the critical cuts as currently defined need not be decisive. Proposition 8b.4, which shows for the standard sketch model that the *exposed* critical cuts are decisive, relies on features being convex. The decisiveness of critical cuts may be salvaged if every curved feature is part of an island which has an inside and an outside, only one of which contains traces, and the curved feature bulges toward that side. In this case one can push Proposition 8b.4 through. Otherwise one must find a new set of decisive cuts using the methods of Section 6D. What's worse, the the rubberband equivalent of a sketch is harder to compute when features are not straight, because Algorithm W no longer applies. Fortunately, one can always fall back to Algorithm F of Section 9B for computing congestion. That algorithm, while slower, is independent of the shapes of features and traces.

10C. The Terminals of Traces

In this section we consider more radical changes to the sketch model than merely altering the definition of what sketches are proper. The changes have two purposes: to allow sketches to represent wiring problems that they previously could not; and to give our sketch algorithms more freedom, so that they may find better and more compact realizations of the sketches given to them. Three extensions of the sketch problems come to mind. One, which was discussed briefly in Section 10A, allows the terminals of a trace to merge during compaction. Another allows terminals to be line segments or convex polygons, and allows trace endpoints to move along the boundaries of these *extended* terminals. The third attempts to remedy the most glaring defect in the sketch model by providing for multiterminal nets: wires that connect to three or more terminals.

All these extensions can, I believe, be incorporated into the sketch model, at the expense of complicating the sketch algorithms and their proofs of correctness. To justify terminal merging and the addition of extended terminals is not too difficult, since these ingredients are already present in the design model. It involves two things: rederiving the correspondence between sketches and designs, and upgrading the sketch algorithms. To add multiterminal nets is extremely hard, however, because it requires a major extension of the design model. In fact, my only reason for thinking that the design model can accommodate multiterminal nets is a strong faith in the proof techniques of Chapters 3 through 7, which have shown themselves in the course of my research to be remarkably adaptable. The main problem is to find the right definitions.

Merging terminals during compaction

Terminal merging was discussed briefly at the end of Section 10A. It begins with the notion that the terminals of each trace in a sketch should not be artificially kept apart; their extents should be allowed to overlap in a proper sketch. We saw how to modify Algorithm T to test routability in the new sense: it must ignore all **degenerate** critical cuts. For if the extents of terminals may overlap, then the sketch routability theorem must be changed to read: *A sketch is routable if and only if its nonempty, nondegenerate, critical cuts are safe.* The definition of degeneracy here corresponds to that in the design model, and may be stated as follows. A bridge β in a sketch S is degenerate if there is a piecewise linear homotopy B such that $B(\cdot,0) = \beta$, for all $t \in (0,1)$ the path $B(\cdot,t)$ is a bridge in S, and $B(\cdot,1)$ is a path in an obstacle of S or in the image of a trace of S and its terminals.

If these changes are adopted, then Algorithm C breaks down in two ways. First, if we retain the provision in the sketch compaction problem that prevents the sketch topology from changing, then the set of acceptable configurations can no longer be represented in as constraint graph. For if some two terminals in different modules can approach arbitrarily closely but cannot coincide, then the set of configurations that represent routable sketches is not closed. Second, protection of the critical cuts is no longer necessary for routability, because a unprotected critical cut can be degenerate and therefore irrelevant. Unlike some changes to the sketch model, this one cannot be accommodated by finding a new sequence of potential cuts with the routability, convexity, ordering, and boundary properties. The problem is that the sketch routability theorem, normally used to justify the routability property, has changed significantly. Fortunately, both breakdowns can be repaired in fairly obvious ways.

We solve the first problem by redefining the configuration space so that the terminals of each trace can merge or even cross over one another. This sort of topological change damages only a small part of the correctness proof of Algorithm C: the claim that the sketches corresponding to different configurations are homeomorphic. We used this claim in Corollary 9d.2 to prove that the adjacency graph of the sketch is independent of configuration, and thus a single adjacency graph could be used for computing flows in all relevant configurations. In fact Algorithm F still computes flow correctly, even when one terminal of a trace passes through the other. For as far as flow is concerned, one may pretend that the first terminal passed above or below the second terminal by a tiny distance.

The second problem may be addressed by changing the definition of protection and reworking the proofs in Sections 9E and 9F. Recall that a configuration $\mathbf{d}$ protects a potential cut ψ unless $\psi(\mathbf{d})$ is an unsafe, nonempty cut. Under the new definition $\mathbf{d}$ protects ψ unless $\psi(\mathbf{d})$ is an unsafe, nonempty, *nondegenerate* cut. The nondegeneracy condition causes no more trouble than the nonemptiness condition; its presence is felt only in Lemma 9e.3.

Only one major change is needed in Algorithm C. All potential cuts must be tested for degeneracy, including the horizontal ones that define the initial constraint set. A straight cut between different modules, which is the only kind Algorithm C ever considers, is degenerate if and only if its trace code matches that of a trace between the same features as the cut. Since there is at most one trace between those features, the test for degeneracy is quick, at least compared to the computation of flow that precedes it. The other change is minor: when preparing the output sketch for routing, Algorithm C should eliminate all traces whose terminals coincide.

Extended terminals

The trouble with sketches is that their features have empty interior. Consequently our proof techniques, which rely heavily on lifting to a covering space, do not apply directly to sketches. All the theorems concerning sketches must be derived from corresponding theorems about designs. But what is a drawback in proving theorems is a virtue in designing algorithms: if all terminals are points, one never need worry where to place the endpoint of a trace. Wl en we relax the restriction that terminals be points, we immediately face several problems in the construction and use of the rubber-band equivalent, and in routing, concerning the placement of trace endpoints. These are the same problems we sidestepped in Chapter 7. Everything I say about routing and testing routability in the presence of extended terminals applies equally well to the design model.

Any convex island in a sketch may be an extended terminal, provided that its trace contacts it from the outside. Thus terminals can be points, line segments, and convex polygons. The restrictions on extended terminals arise because the terminals of a wire in a design are convex, inner fringes.

Because extended terminals are an integral part of the design model, the correspondence between sketches and designs can easily admit them. A realization of a trace is any bridge-homotopic trace, and so one may move the endpoints of a trace along their respective terminals. Thus the notion of homotopy for traces is in line with that for wires. The sketch routability and routing theorems go through essentially as in Chapter 8, which is to say, with either a lot of handwaving or a lot of hard work. (Note: I have not actually done the hard work, so there is some chance that the extension contains a fatal flaw.) One difference is that the ideal realization of a trace is no longer necessarily unique.

Routing is also more difficult when extended terminals are present, due to the need to locate trace endpoints. I expect, however, that the following approach can be made to work. As in Algorithm R one first computes for every trace a corridor for each diagonal slope. But now these corridors should include doorways that pass right through the trace's terminals. The partial realizations of the trace are now the shortest paths through these corridors from one terminal to the other. Because terminals are convex, the partial realizations should not be too difficult to compute. Now one merges the partial realizations as before, with some minor extensions to determine from which partial realizations the ideal realization takes its endpoints.

Detecting trivial crossings

The other things that must change when extended terminals are present are the algorithms for computing congestion and necessary crossings. In particular, the rubber-band equivalent of a sketch must be treated somewhat differently. As explained in Section 7C, every crossing between a straight cut and a rubber band is either necessary or *trivial*. Informally speaking, a crossing is trivial if one of its corresponding half-cuts is homotopic to a path in a single island. See Figure 10c-1. The trivial crossings used only to occur at trace endpoints, but that is no longer true. Hence the RBE itself can no longer identify the trivial crossings; some additional computation is needed.

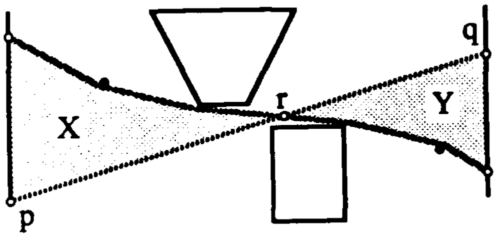


Figure 10c-1. *Trivial crossings.* **The crossing at r between the cut $\overline{pq}$ and the grey trace is trivial if and only if one of the regions X and Y is empty of features. If the cut and trace shared only one terminal, one of these regions would be absent.**

Fortunately, we have considerable freedom in choosing where on its terminals a rubber band should begin and end. Technically, the rubber band for a trace is the shortest path through a certain corridor that begins and ends at the endpoints of the trace. But we may replace the rubber band of a trace by the rubber band of any route for that trace without affecting the content of any cut. For starters we choose each rubber band so that neither its first nor its last segment lies within a terminal. Under this condition only the first and last crossings of a cut can be trivial.

A crossing may be tested for triviality as follows. For each terminal shared by the cut and the rubber band, consider the loop formed by the terminal and the portions of the cut and rubber band between the terminal and the crossing. If the inside of this loop is free of features, then the crossing is trivial. But if every such loop encloses a feature, then the crossing is nontrivial, and therefore necessary. If the trace and the terminal share no terminals, then nontriviality is automatic.

Testing whether a loop encloses some feature is generally difficult, but can be simplified by a judicious choice of rubber bands. We say that a rubber band ρ for a trace θ is **stiff** if no subpath of ρ that is not straight can be replaced by a straight path to yield the rubber band of a route for θ. Every trace has some route whose rubber band is stiff, because one can keep eliminating joints of the rubber band until no further subpath can be straightened. For a crossing of a cut by a stiff rubber band ρ to be trivial, it must occur in the first or last segment of ρ; otherwise one could straighten out a subpath of ρ. Hence if stiff rubber bands are used, the loops we must test are essentially triangles. One can then apply any of the various retrieval or counting algorithms for triangles [7, 56] to test triviality of crossings. Good average-case performance, say $O(\log n)$ or $O(\log^2 n)$ per search, can probably be achieved with a quadtree structure.

Stiff rubber bands may be computed according to the following outline. One first uses Algorithm W to compute the "envelope" of the rubber bands of routes of a given trace, as shown in Figure 10c-2(a). The left-hand boundary is a rubber band that begins so far counterclockwise on the first terminal that its first segment lies on that terminal, and ends so far clockwise on the second terminal that its last segment lies on that terminal. The right-hand boundary is similar. If the two boundaries of this envelope do not intersect, then a straight rubber band exists. If they do intersect, then they intersect along a path that forms part of the middle of the desired rubber band. This path need only be augmented by straight paths from the first terminal and to the last terminal. If possible, these straight paths should be collinear with the segments to which they attach. See Figure 10c-2(b). I omit the details because I believe there are better ways of computing necessary crossings and congestion in the presence of extended terminals. One candidate method is mentioned in the following section.

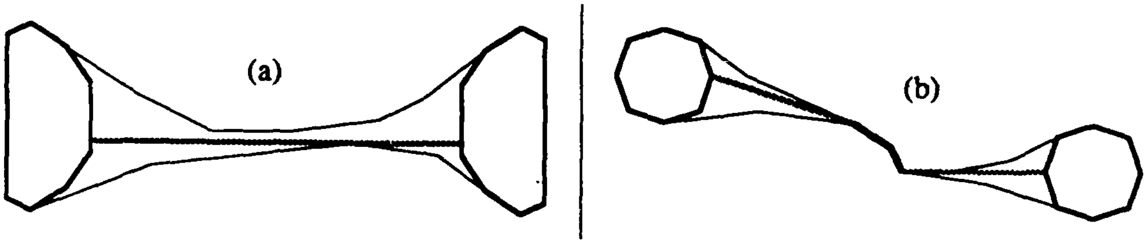


Figure 10c-2. *The envelope for a rubber band.* In (a) the boundaries are separate, and the desired rubber band (grey) is straight. In (b) the two boundaries intersect in the middle (black segments) of the stiff rubber band.

Algorithm F of Section 9B needs modifications along the same lines. Like the RBE, Algorithm F essentially computes a minimal sequence of crossings of a cut, and if the endpoints of that cut lie on extended terminals, then the first and last crossings may be unnecessary, i.e., trivial. To test whether a crossing of a trace by a cut is trivial, one looks at each extended terminal they share, and compares the trace codes (gate lists) of the portions of the cut and trace from that terminal up to the crossing. If they are equal, then the crossing is trivial. To incorporate this test into the optimized version of Algorithm F is not trivial, but it can be done.

Multiterminal nets

Perhaps the most problematic weakness in the sketch model is its complete in-ability to represent wires with more than two terminals. I now present an extension of the sketch model that may alleviate this problem. Of course, one can always break a multiterminal wire into two-terminal wires by introducing connector modules (groups of terminals) along the way. But doing so defeats the purpose of having flexible interconnections.

We may represent a multiterminal wire by a ring-shaped set of traces, as shown in Figure 10c-3(a). I call this set of traces a **net**. The traces in a net can intersect, and indeed they must intersect at their terminals unless extended terminals are present. But no two traces in the net may make a necessary crossing, so Figure 10c-3(b) is ruled out; and the loop that the traces form must enclose no features. Furthermore, the traces in a net must have the same width, as must the terminals of the net, and the width of the traces may not exceed that of the terminals. To route a net is to route each of its traces; the result is always a net. For technical convenience we assume that two-terminal wires, as well as wires with more terminals, are represented as nets. (A two-terminal net is just a pair of bridge-homotopic traces.)

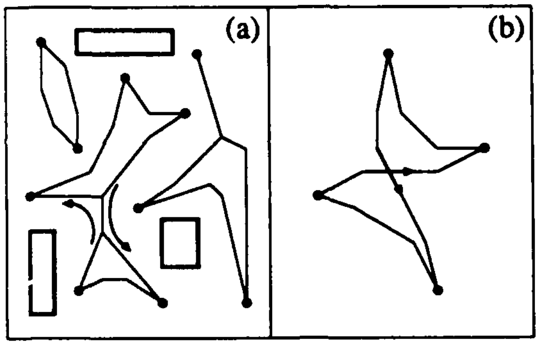


Figure 10c-3. *Valid and invalid multiterminal nets.* Part (a) shows a sketch with nets in place of traces. Each net is a loop, not necessarily simple, of traces (light paths), no two of which cross over. The paths in part (b) therefore do not form a net.

A net need not represent the final form of a multiterminal wire. Instead, having routed the nets in a sketch, one can then replace them by more conventional tree-shaped wires. Each wire's centerline should be placed within the net and the points it encloses. Unlike the traces of the net, however, this tree-shaped wire will not generally have its total arc length minimized; to minimize it involves solving a Steiner tree problem, which is NP-complete [13].

There are two fairly natural separation rules one might apply to nets, one stricter and one looser. Both insist that in a proper sketch no trace may cross over another, although two traces in the same net may coincide along part of their length. The strict rule treats the traces in a net as independent entities, each with its own extent; it says that a sketch is proper only if whenever two of its elements (traces and islands) have overlapping extents, they are either a trace and one of its terminals, or two traces that share a terminal. It further requires that each trace in a proper sketch be self-avoiding. The other is more akin to the rule for terminal merging; it treats each net together with its terminals as a whole object. We define the extent of a net to be the union of the extents of its traces and terminals. Under the loose rule a sketch is proper only if the extents of its nets and obstacles (nonterminal islands) are disjoint and its nets are self-avoiding, meaning that no net has an extent that separates two islands of the sketch.

Unfortunately, only the strict rule is likely to give rise to a satisfactory routability theorem, and so we adopt this one. Under the loose rule there is no satisfactory definition of congestion. (Making sense of the loose rule requires allowing each trace to pass over terminals in its net, which the present framework absolutely forbids.) The conjunction of terminal merging and multiterminal nets must wait for the model presented in Section 10D. Under the strict rule, the congestion of a cut $\overline{pq}$ may be defined as the total width of the traces in the content of $\overline{pq}$ after eliminating every second trace wherever consecutive traces in the content are part of the same net. One must remove alternate traces because when a cut necessarily crosses a net, it usually makes necessary crossings with two traces of the net—one entering the inside of the net, and one leaving. Exceptions occur only at the endpoints of the cut, and then only if those endpoints are terminals.

Impact of multiterminal nets

Provided that all the mathematics works out, the sketch routing and routability theorems should continue to hold, and the sketch algorithms should change only slightly. The only major changes come in the computation of congestion by Algorithm F and the rubber-band equivalent, and the computation of doorways by Algorithm R. What changes in Algorithm R, of course, is that pairs of crossings by traces in the same net must be considered as single crossings for the purpose of computing the lengths of struts. Algorithm T now requires that each cable in the condensed RBE be assigned a width that reflects the total width of adjacent pairs of strands belonging to the same net, and must also record any strands left over so that Algorithm T can correct for duplication of crossings. Finally, when Algorithm F computes the content of a cut, it must also collapse pairs of consecutive traces when they fall in the same net. (Section 10A described similar modifications to Algorithms T and F. Like these, they arose from a definition of congestion in terms of content.)

The hard part, of course, is proving that the sketch algorithms remain correct. I see no other option than to extend the design model and generalize the whole theory of single-layer routing. If one stuck to the same outline, at the very least Chapters 4 through 6 would have to be overhauled. In discussing other extensions of sketch problems I have expressed confidence that the relevant theorems could be strengthened to match, but here I cannot. Many of the proofs in these chapters use the assumption that a terminal intersects at most one wire. To eliminate this assumption in the presence of the new definitions may be easy, or it may be impossible. Whether the sketch model can accept multiterminal nets is really an open question.

10D. An Alternative to the Sketch Model

No discussion of sketches would be complete without a mention of alternative models. We begin by exploring the problems involved in representing how wires connect to their terminals. The difficulty of adding extended terminals and multiterminal nets to the sketch model, and the awkwardness of working mathematically with sketches, suggest that an entirely new model may be needed if our understanding of single-layer wire routing is to be advanced. In this section I present a pair of models for single-layer routing problems that may resolve these two issues. One, like the design model, is designed for mathematical convenience, while the other, like the sketch model, is intended for algorithmic use. By offering a new perspective on the connection of wires to their terminals, they handle extended terminals and multiterminal nets in an elegant manner. And because the two models are closer together than sketches and designs, they promise a smoother connection between the mathematical and algorithmic parts of the theory.

Modeling of terminals

The connection of wires to their terminals is the major stumbling block in the development of general models for single-layer routing. (A glance at the topics of the preceding section may help to convince you.) Terminals cause difficulties both in the technical development of the model and in its use. Because a wire has terminals, its endpoints must be treated differently from its middle, and its terminals must be treated differently fiom all other obstacles. For instance, because of the special status of terminals, the rubber-band equivalent of a sketch must distinguish between trivial and nontrivial crossings. This problem is particularly acute when extended terminals are present. In the design model, the burden of keeping track of respect and degeneracy for cuts and half-cuts can be traced to the possibility of a cut or wire winding around a terminal.

Slight changes in the way terminals are managed can make or break routability theorems and routing algorithms. As an example, suppose that we allowed a sketch to have extended terminals but fixed the endpoints of each trace at specific points on those terminals. To make things more plausible, we may permit the traces to run along their terminals for some distance before departing. With this extension, my theory of routing collapses. Consider a trace that happens to wind once or more about one of its terminals. The middle of this trace can come arbitrarily close to its (fixed) endpoints, but cannot intersect them. Hence we must we give up all hope of routing with minimum-length traces. Moreover, we can no longer expect to decide routability by the safety of certain cuts; some cuts will need capacities that strictly exceed their congestions. In other words, some of the routability conditions will go *from closed to open.* One can alleviate these problems by allowing traces to have self-intersections. The result is a model that is more complicated than the original without being any more expressive.

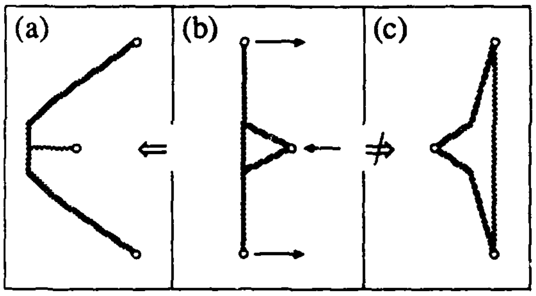


Figure 10d-1. *Compaction with multiterminal nets.* Suppose that nets are added to the sketch model, and suppose that during compaction the terminals in the sketch part shown in (b) moved in the directions of the arrows. The net could not move into the desirable configuration (c), but would be forced into something like (a).

The problems of terminal connections come to the fore when multiterminal nets are considered. Many seemingly natural ways of handling multiterminal wires simply do not work. One try that fails is the "loose" wiring rule described at the end of Section 10C. The stricter rule that we adopted is not an entirely satisfactory foundation for a study of routability either. It disallows routings that might often be desirable, by a prohibiting a trace to approach the terminals in its net except those to which it connects. Figure 10d-1 is only the simplest example. During routing or compaction one might greatly improve the layout by moving part of a net across one of its terminals. But this sort of topological change is foreign to the sketch and design models.

The network model

Now we come to the point of this section: a novel perspective on wire/terminal connections that gives rise to very pretty alternatives to sketches and designs. For concreteness I discuss the idea as an modification of the sketch model, and call the analogue of a sketch a **network**. We think of a wire as a **net**, a simple loop in the routing region that *encloses* its terminals rather than intersecting them. See Figure 10d-2. A net may not touch any terminal or obstacle, but it must enclose at least one terminal and may enclose more than two. Terminals may be islands of any shape. No two nets may intersect, and none may enclose another. To route a net, we replace it by any other net that is homotopic as a map of the circle S^1 into the routing region.*

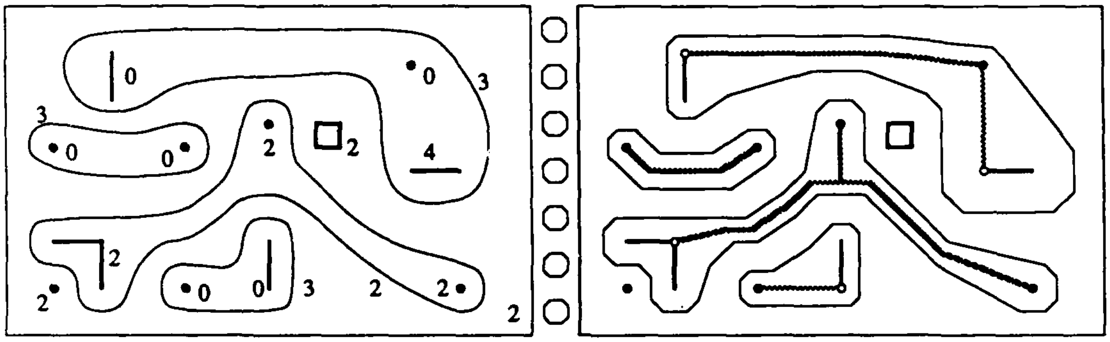


Figure 10d-2. *The network model.* At left is a network of islands (made of dark points and lines) and nets (light curves). Each of these elements has a width, the number next to it. Between the two layouts are several copies of the unit polygon. At right is the unique proper realization of this network with minimum-length nets. Within these nets one can route the centerlines (grey lines) of multiterminal *wires* that are properly separated.

The design rules for nets are similar to those for sketches and designs, but are perhaps even simpler. Each net has a positive width and a corresponding territory, the set of points whose distance from its image is less than half its width. In the simplest case islands have zero width, and their territories are just themselves. A layout is proper if and only if its elements (nets and obstacles) have disjoint extents and its nets are self-avoiding. Thus the terminals of a net are treated no differently than any other obstacles. A net is self-avoiding if the complement of its extent has only two components that contain obstacles: one inside the net, and one outside. If desired, islands may be given positive widths and corresponding territories. The widths of a net's terminals need have no special relation to one another or to the width of the net.

I conjecture that the sketch routability and routing theorems carry over to this "network model" in an obvious way. Cuts, capacity, safety, and emptiness remain

* There is an amusing parallel between my models for multiterminal connections and the models once discussed by particle physicists of the confinement of quarks in various subatomic particles. According to current theory, each nucleon and each meson is made of quarks bound together by a force that increases with distance, so that no quark can be isolated from the others. For the purpose of predicting properties of these particles, two models of quark confinement were introduced: a "string model", which pictured the quarks as being held together by elastic strings, and a "bag model", which represented the binding force as a flexible bag containing the quarks. One hears less of these models now, presumably because quark interactions have become better understood.

the same, as do the critical cuts. One must replace 'trace' by 'net' in the definition of congestion. Then the routability theorem reads: *A network is routable if and only if its nonempty critical cuts are safe.* The routing theorem remains at full strength: *In a routable network, every net has a unique minimum-length feasible realization, and these realizations form a proper network.* In other words, net lengths can be simultaneously minimized, and the layout that does this is unique. One can even modify the network model to permit the extents of the terminals of each net to overlap. As with sketches, the only change is to eliminate **degenerate** cuts from those that determine routability. Here a cut is degenerate if and only if (a) it can be collapsed into an island, or (b) some bridge-homotopic cut is entirely enclosed by a net.

Aside from the theorems it may support, the network model has two very nice properties in itself. First, it contains within it an isomorphic copy of the sketch model. To convert a sketch into a network, replace each trace by a net that surrounds its terminals, give that net half the width of the trace, and deduce the width of the trace from the widths of its terminals. I claim that the resulting network is routable if and only if the sketch was routable. This claim is not even very hard to prove. Second, the network model needs only minor changes to relate it to designs instead of sketches. One needs only replace the routing region by a sheet, and change some terminology: islands become fringes, territories become extents, and so on. The notion of net homotopy, in particular, needs no change. In fact, the only substantive change is that the routing obstacles grow to become polygons; and these were already allowed. So by mildly restricting the obstacles in the network model, we obtain a model that can be analyzed using covering spaces.

Benefits of nets

The network models may have many applications to single-layer wire routing, but it arrived too late in my research to have any influence on the bulk of this thesis. So close are the two network models to one another, and so naturally are the sketch and design models embedded in them, that they may actually form a better bridge between sketches and designs than the direct correspondence of Chapter 8. Even if not, the idea of converting wires and traces into nets could be of significant use in dealing with extended terminals. See Figure 10d-3. When computing flow, the first and last crossings of a cut would be the only trivial ones, and so no complicated test for triviality would be required. (This idea works only for two-terminal wires.)

But the network models are really intended to support a new and better theory of single-layer wire routing, all the way from basic topology to algorithms. If such a theory could be developed, it would have several advantages over that presented here.

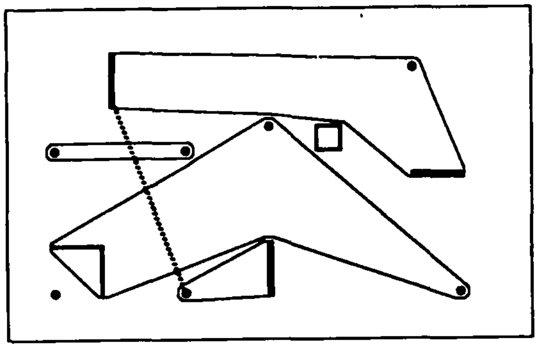


Figure 10d-3. *The rubber-band equivalent of a network.* The loops shown here are the rubber bands for the nets in Figure 10d-2. The congestion of a cut equals the sum, over all crossings of that cut by rubber bands, of the width of the net corresponding to that crossing.

- It would treat all nets as fully flexible interconnections, regardless of the number of terminals they enclosed. In particular, it would solve the problem of Figure 10d-1: as terminals move or other obstacles intrude, each net deforms so as to connect its terminals in the best way.
- It would neatly decompose the routing problem for multiterminal wires and wires with nonconvex terminals. For such wires the problem of minimizing total arc length is generally difficult. When the net corresponding to a wire was routed, it would delimit the region in which that wire should run. Any reasonable heuristic could then be used to route the actual wire. In the case of nets with two convex terminals, the heuristic could be replaced by a fast algorithm that minimizes wire length.
- It would handle extended terminals with no penalty in efficiency or algorithm complexity. The problems of placing wire and cut endpoints would vanish, taking with them the need to distinguish path plans and link plans. Related technical difficulties, like the possibility that consecutive gates in an ideal wire's tunnel intersect, would also disappear.
- The RBE data structure for computing flows and crossing sequences would become simpler. Every net has a unique rubber band, regardless of terminal shape, and every crossing of this rubber band by a cut corresponds to a necessary crossing of the net by the cut. Compare Figure 10d-3.
- The concept of respect would disappear entirely from the theory, thus simplifying many proofs. Degeneracy, also, would play a much smaller role, except where terminal merging is concerned.
- Finally, the task of relating the two network models would be much simpler than the task of relating sketches to designs. In particular, the algorithms for routing, testing routability, and compaction would be nearly identical in the two models.

In contrast, I can think of only a couple of disadvantages that would accrue to algorithms and theory in the network model. One is that the algorithms would be slightly slower in the case of two-point nets, since there would be approximately twice as many crossings between nets and cuts as between traces and cuts. The path-finding algorithms might also need to run in two passes because they cannot initially identify any point that the output net should pass through. I am confident that no significant new algorithmic ideas would be needed. The only telling objection is mathematical. Because nets are essential loops, they cannot be lifted to a simply connected covering space of the routing region. A different covering space is needed for analytical purposes: the universal covering space among those to which all the nets can be lifted. Its properties are harder to derive than those of blankets. Consequently, the theory of the network model may depend on more advanced topological concepts than those I have employed.

Conclusion

A Critical Review

This dissertation did not begin the study of wire routing with homotopy constraints, and neither can it end it. At the risk of exhausting the reader's patience, therefore, I will say a few more words about the sketch problems. The first part of this Conclusion summarizes my main results on routing and compaction, puts them in perspective against the practical problems of wire routing, and then recognizes several related works, including several from which my papers have drawn and one to which my research has already contributed. The second part takes a careful look at the sketch model, and draws from that look several suggestions for further research.

A. Summary of Results

I have presented efficient algorithms for three problems of single-layer wire routing with homotopy constraints: *sketch routability*, *sketch routing*, and *one-dimensional sketch compaction*. These problems are natural abstractions of placement and routing tasks that circuit designers face when they distinguish flexible wires from rigid features. The tasks are these: determine whether a layout is routable under a given topology and layer assignment; if so, route it with wires of minimum length; and compact the layout horizontally, introducing jogs into wires in an optimal fashion. My solutions to the sketch problems work in a variety of wiring models, and they can be extended to handle a variety of useful constructs.

With the possible exception of the sketch compaction algorithm, all the algorithms are efficient enough to be used in practice. The algorithms for sketch routing and routability testing run in time $O(n^2 \log n)$ on input of size n. Sections 1E and 1F suggest that their average-case time requirements can be reduced to $O(n^{3/2} \log n)$ time or less. The sketch compaction algorithm runs in time $O(n^4)$, while its average-case performance is probably $O(n^3 \log^2 n)$ or better.

My algorithms employ two important data structures that help in computing the properties of cuts. One—the *rubber-band equivalent* of a sketch—is geometric, and applies only to straight cuts. It supports fast computation of crossing sequences and congestion via scanning. It handles straight cuts only. The other—the *adjacency graph* of a sketch—is topological, and handles arbitrary cuts. It supports somewhat slower computation of congestion via graph search.

To justify and explain the sketch algorithms, I have developed a body of definitions and theorems that I refer to (somewhat pretentiously) as a theory of single-layer wire routing. In reality it is a partial theory of two particular models of single-layer routing: the sketch model and the design model. The centerpieces of the theory are two theorems that characterize the routability of a design and the optimal routing of a routable design in terms of the attributes of certain cuts and half-cuts. The *design routability theorem* states that a design is routable if and only if its major straight cuts are safe. The *design routing theorem* shows that every routable design can be routed so as to minimize the length of every wire, and it characterizes the optimal routing of each wire as the minimum-length route of that wire whose nontrivial straight half-cuts are safe. My proofs of these theorems give the first mathematically sound foundation to single-layer wire routing in multiply connected regions. I have also shown how to carry the design theorems over to the sketch model, and thereby provide the first rigorous treatment of algorithms for wire routing with homotopy constraints.

My treatment of the design model introduces a powerful tool for analyzing one-layer routing: lifting to a *covering space of the routing region*. None of the problems I encountered in studying the design model required me to step outside this framework. Covering spaces allow one to formalize and study many deceptively simple-looking concepts that play key roles in the routing problem. Some of these notions—flow, necessary crossings, and barriers—made sense when routing in simply connected regions, and via covering spaces can also be applied to designs. Others, such as the relations of respect and degeneracy between cuts and designs, have no such counterparts, and yet to overlook them can lead to disaster. (Before I hit upon the right definitions for these concepts, they caused persistent problems.) The use of covering spaces also allows one to treat cuts that are not simple, which opens the possibility of analyzing half-cuts in terms of their associated cuts.

Applications to practical problems

After all the algorithms are presented, analyzed, justified, and extended, the question of their practical utility remains. Not having implemented them in any form, I cannot answer with certainty. Nevertheless a short reply is possible. The sketch algorithms are limited in their practical applications mainly by the ability of sketches to represent active devices in integrated circuits. Where design rules are simple, the sketch algorithms—or algorithms derived from the same ideas—show promise. Compared to modeling, performance is less of an issue. I have presented several ideas for improving the average-case behavior of the sketch algorithms, and there are undoubtedly many more to be found by implementing and experimenting with them. Moreover, the constants in their time and space bounds are small.

Most practical routing problems involve multiple layers, and do not specify the topology of the routing. I have little to say about such problems; my results concern provably efficient (polynomial-time) algorithms, while the problems of greatest practical interest are all NP-complete. Nonetheless, there are several ways in which my results might be applied to multilayer routing. Most obviously, a multilayer routing problem can be reduced to single-layer problems by first choosing rough routings of wires, and then assigning them to layers and placing the vias where they change layers. Good heuristics are known for the first problem, which is called *global routing*. Unfortunately, few heuristics are known for layer assignment and via placement (but see [41]).

A more robust approach allows for local topological changes, such as moving a wire across an obstacle or another wire. Starting with an infeasible assignment of wires to layers, one can identify the unsafe cuts, and try to shift wires away from them to reduce their congestion. This process is facilitated if on each layer the wires run in a preferred direction, as is common in integrated circuits and multilayer printed circuit boards. At least one PCB router was implemented using such a technique [9]. Its designer noticed that its "heuristic" method of testing whether a layer was routable, namely checking that no cut had greater congestion than capacity, never seemed to err. This suggestive piece of experimental evidence provided the initial motivation for my work.

Although this dissertation emphasizes routing, my compaction algorithm is considerably more powerful than my routing algorithm, and is more likely to prove useful. The reason is simple. In compaction, one may reasonably assume that the topology of the layout and the layer assignment are given, while in routing, most of the problem lies in choosing a proper topology and layer assignment. This observation makes the theory of routing no less important, however, for the compaction algorithm uses the routing algorithm as a subroutine.

The sketch compaction algorithm may also have applications to routing problems. The idea, which has been tested on channel routing with excellent results (see Acknowledgements, in the Preface), is as follows. One first expands the layout so that a conventional routing program, which may have difficulty with crowded layouts, can succeed. One then applies a compactor with the ability to insert arbitrary jogs in order to compact the layout to its proper size.

Related work

This thesis synthesizes and generalizes results from three primary sources. One is the paper by Tompa [52] that solves the problem of river routing in a rectangular channel with terminals along its top and bottom. This paper first introduced the notion of the *barriers* for a wire. It showed that every routable channel can be routed by choosing a minimum-length barrier-avoiding routing for each wire. That demonstration formed the outline for my proof that the ideal embedding of a design is safe. Another source is the algorithm of Leiserson and Pinter [22] that uses routability conditions to compact a river routing channel horizontally. The third source is the paper of Cole and Siegel [6] that solves Pinter's problem called 'DRH' [41], which is essentially the sketch routability problem in the grid model. That paper first claimed the equivalence of routability and safety—what I would call the sketch routability theorem for grids—but without a detailed proof. I relied upon that result in an earlier paper [21] when my own attempts at proving the routability theorem were failing, but my present proof is independent of their result. The algorithms and theorems in this thesis subsume those in the sources just mentioned, but of course the special-case algorithms are faster and easier to prove correct.

Recently Storb et al. [33] have developed an algorithm that appears to solve the sketch routing problem in an arbitrary wiring norm. Building on the algorithms for constructing the rubber-band equivalent of a sketch and for testing routability, they propose a routing algorithm for the euclidean wiring norm with a worst-case execution time of $O(n^3)$. If this result is borne out, it will complement my routing algorithm, which is faster (time $O(n^2 \log n)$) for polygonal wiring norms and slower (time $O(n^4 \log n)$) for other norms. Their method involves sweeping through a with scan lines perpendicular to each rubber band, constructing barriers and routing through them as in [52]. They, like I, call upon a simply connected covering space to understand how separate parts of a wire interact. How difficult their algorithm is to implement, how complicated a correctness proof it will need, and how quickly it can solve problems of practical size—the answers to these questions are as yet unknown. The same authors point out that if the input to the routing problem includes not only rough routings of the wires, but also a planar graph in which their realizations must run, then the problem of routing with minimum total wire length becomes NP-complete. (Wire length can be minimized in certain fixed graphs, such as grids.)

All the works I have just described refer explicity or implicitly to the dependence of routability on the congestions and capacities of cuts. This phenomenon occurs in other routing problems as well, notably the problem of routing edge-disjoint paths in a planar graph between pairs of terminals on its outer face [3, 17, 32]. The algorithms for this problem and its various special cases are all derived from a theorem [40] concerning the existence of such paths. That theorem states that there exist edge-disjoint paths connecting the terminals if the *free capacity* (margin) of every cut is nonnegative and even. (Here a cut is not a path, but rather a partition of the vertices into two sets.) This routing problem is rather different in character from mine, however, because it allows paths to cross.

B. Directions for Future Research

I close with a critical look at my models and a discussion of open problems. The deficiencies of the sketch model, in particular, suggest several directions for further investigation. My conclusion is that a great deal of work remains to be done in all the areas I have touched upon—the application of topology to prove routing theorems, the design of efficient algorithms and data structures, and the implementation of those algorithms.

A critique of the sketch model...

The many extensions presented in Chapter 10 may convince some readers that the sketch model is a robust one, and in some ways it is. But the reader who has looked carefully at the amount of work needed to justify the extensions may come to a different conclusion. Indeed, one might question whether my treatment of the unadorned sketch model is adequate, given the amount of handwaving in Section 8C. All my results concerning sketches rest not only on the detailed theory of the design model, but also on a complicated limiting process that relates this model to the sketch model. Three of the extensions I have proposed—elements of differing materials, curvilinear sketch elements, and multiterminal nets—involve strengthening both supports of the sketch results. What's worse, there is no sure way to tell whether these extensions would somehow interfere, except by carrying through the proofs in as general a model as possible. Though generalizing a theory is usually much easier than constructing it from scratch, in this case the sheer mass of technical material makes it a daunting task. Still, in my opinion there is no substitute for a rigorous correctness proof of an algorithm, except perhaps an extensively tested, practical implementation.

But there are troubles on the practical side as well. Because sketches are so abstract, there is no straightforward way to convert a more conventional representation of a integrated circuit layer into a sketch for the purpose of routing or compaction (although the reverse is easy). On the other hand, even with all envisioned extensions, sketches may be too simple to serve as the primary geometric denotation of circuit layers in a CAD system. Even if they could, their ability to represent devices and multiterminal nets is disturbingly inflexible. We saw the difficulty of representing transistors in Section 10A, and the problems with multiterminal wires were made apparent in Section 10D. Sketches make more sense as an abstraction of the interconnections among large circuit blocks than of the wiring within "leaf cells".

To be fair, the sketch abstraction may be perfectly adequate for printed circuit board layers. The components of a PCB layer, terminals (vias) and traces, are easily identified and correspond directly to sketch elements. More permissive wiring rules than grid-based rules are common, and fit nicely into the sketch model. Finally, optimal treatment of multiterminal nets may not be crucial. But except for the sketch routability theorem and its implications for routability testing, the application of my results to circuit boards is not particularly interesting. The most powerful sketch algorithm is Algorithm C, and it would find little use in compacting circuit board layouts, since modules (in this case, chips) are not generally free to move. In any case, their placement is dictated more by the physical volume they occupy above the routing layers than by routability constraints.

...and an apologia

So the sketch model sits in an awkward position. It is more abstract and simplistic than a practical user would like, yet it is mathematically inconvenient. For these reasons I have come to regard the sketch model as deficient. Why, then, did I choose it? The reasons are mainly historical. It was a natural outgrowth of the grid models previously in vogue among theorists, and I had previously published algorithms using it [21, 28]. These were the algorithms I set out to justify. I could have taken up these algorithms in the design model, for example, rather than the sketch model. In a sense the two models are on equal footing: while the design model is flawed for representing real designs, the sketch model flawed for mathematical analysis. But because the terminals in a sketch are points, the algorithms are simpler in the sketch model, and this fact tipped the balance. The algorithms dictated the model, not the other way around.

Of course, none of the flaws in the sketch model means that study of sketches cannot provide valuable insight into the issues of routing, routability testing, and compaction in more complex models. But if insight is all that we carry away, why spend so much time analyzing the technical aspects of one model? The answer is simply that some model must be chosen; the technical details come with the territory. Surely there are models better suited for practical use. But building a practical system was never an objective of my research. The real objective was to build technical tools, where none were previously known, for solving problems of single-layer routing, routability testing, and compaction in the presence of homotopy constraints. This I have done by developing the theory of routing in the design model, and showing how results from one model can be brought over to another.

Directions for future research

One way to build upon the ideas in this thesis would be to redevelop them in a model that avoids some of the defects of the sketch model. The sketch model is weak in two general areas: representing devices and modules in integrated circuits, and representing wiring structures like multiterminal nets and extended terminals. I see two corresponding directions for model changes: toward greater accuracy and faithfulness in representing practical circuits, and toward greater versatility and mathematical cleanliness. Regarding cleanliness, I have no solution to the problem that the use of covering spaces requires features to have positive diameter, which precludes the attachment of wires to terminals of the same size. Consequently, I see no good way to analyze the models I am about to discuss except by passing to an auxiliary model (like the design model) whose routing region is a manifold, and carrying back the results via convergence arguments. But we can hope to make this process easier, as in the network models of Section 10D.

As we noted in Section 10A, the sketch model has trouble representing the variety of different materials that interact on the lowest layers of a chip. The root of the problem is that the only objects that can coexist or coincide in a sketch are traces and their terminals. A real chip has many types of elements whose extents can overlap, not just wires and terminals. Typically a wire can overlap or approach regions in the device to which it is connected, but wires that are not connected to that device must stay away. Several good ideas for representing wires and devices in a simple and quickly accessible form may be found in [37]. Some of them might be incorporated into the sketch model. Most useful would be a way of relaxing the separation constraints among the parts of a device and the wires that connect to it.

Another possibility is to return to a view of modules as polygons with terminals on their boundaries. Point modules (isolated terminals) should also be permitted. One would have to confront directly the complicated problem of module interconnection: how to represent preexisting cells of a design in a compact form that facilitates routing and checking of design rules among modules. An advantage of solid modules over collections of scattered obstacles and terminals is that they can hide what might be design-rule violations in the context of routing, but are actually proper due to the function of the device. Module boundaries could include pointlike terminals, extended terminals, and other features. The layout should assign materials to all wires and features should be assigned materials, and could potentially mandate a different separation constraint between each pair of materials.

Two main technical issues would arise in a model based on modules: what paths should be considered cuts—paths landing too close to a terminal would not qualify—and what restrictions must be placed on the separation distances and the composition of module boundaries. The goal, of course, would be to prove routability and routing theorems like those for sketches, and adapt the sketch algorithms to the new model. I contend that the concepts developed in the design model will be useful in new models as well. In particular, congestion can be measured by relating it to a flow-like quantity defined using covering spaces.

Open problems

In addition to the general problem of finding better models for single-layer wire routing, there are several specific questions that future research could aim to answer. Those concerning extensions of the sketch problems were discussed in Chapter 10; I list the most significant of them below.

- Can a single sketch incorporate wires of differing materials, and hence differing separation requirements?
- Can the sketch theorems and algorithms be enriched to provide for extended terminals and multiterminal nets?
- Does the network model (see Section 10D) support efficient algorithms for routing, routability testing, and compaction?

I conjecture that the answer is yes in each case. The remaining questions are larger and harder, and I make no predictions about them.

- How efficiently can the sketch routing problem be solved when the wiring norm is not polygonal?
- How fast can the sketch routing and compaction algorithms be made to run on practical examples? In what cases are they superior to algorithms that treat wires as objects to be moved?
- Can Algorithm C be extended to handle an unroutable initial configuration? How should the sketch compaction problem be defined in this case?
- How can routability constraints be applied to two-dimensional layout compaction? What data structures might be used for computing congestion when features were moving in all directions?
- What mathematical tools can assist the study of the network model? Does this models indeed support strong routability and routing theorems?

And for the mathematically adventurous, there is the following question:

- Can the results of the design model be generalized to higher dimensions? (Can one route surfaces in R^3 among toroidal terminals and obstacles?)

Further studies of routing problems with homotopy constraints, even those with little or no relevance to practice, may prove fruitful by clarifying the general principles at work.

Glossary

Since this thesis spans three areas that are only weakly related—topology, algorithms, and circuit layout—and introduces a substantial amount of new terminology, I have collected here the definitions of terms that are likely to be unfamiliar. Due to software limitations, however, I have not included pointers to the places where the terms are first used.

absolute retract: A space A is an absolute retract if whenever a normal space X has a closed subspace B homeomorphic to A, then B is a retract of X. The fact that I and R^1 are absolute retracts follows from the Tietze Extension Theorem [38, p. 212].

adjacency graph: A data structure used by Algorithm C, the sketch compaction algorithm, for computing congestions of straight cuts. See Sections 7C and 9B.

akin: Subcuts are akin if their liftings connect terminals in the same way. See Definition 4d.1. Subcuts that are akin have link-homotopic associated cuts, and therefore share properties such as respect for a given design. Two crossings of links (or chains for links) are akin if those links can be lifted to reflect the crossings such that corresponding liftings share their terminals. Two plans (crossing sequences) are akin if they have the same length and corresponding crossings are akin.

angle: In Sections 7Dand 7E, a point of the unit polygon of the wiring norm. The angle at which a path σ travels is the normalization of the vector $\sigma(1) - \sigma(0)$, assuming that σ is not a loop.

arc length: The arc length of a path α in the norm $\|\cdot\|$ is defined as follows. If α is piecewise linear, then its arc length is the sum over all its segments τ of the quantity $\|\tau(1) - \tau(0)\|$. Otherwise the arc length of α is the supremum of the arc lengths of the polygonal approximations to α, that is, piecewise linear paths β from $\alpha(0)$ to $\alpha(1)$ such that $\alpha(s) = \beta(s)$ for each joint s of β. By default we measure arc length in the *euclidean* norm.

arrangement: An arrangement on a sheet S is a finite set of disjoint simple cuts in S.

article: A connected set of *details* in a design: either a nonterminal fringe, or the image of a wire together with the wire's terminals.

aspect ratio: The ratio of a rectangle's longer dimension to its shorter dimension.

associated cut: A cut formed from a half-cut or mid-cut by (1) extending it along its link(s) to form a link, and (2) applying a link homotopy to obtain a cut. In large measure, the associated cuts determine the properties of a half-cut or mid-cut.

barrier: In general, a barrier for a wire is a connected area that no *feasible* realization of the wire can enter. In the design model, a barrier is a subset of a *forbidden zone* that constrains the lifting of a wire, rather than constraining the wire directly. See Section 5A.

base: A fringe that contains the set from which a *barrier* grows (cf. Lemma 5a.4). Also, the range of a *covering map*.

base point: A distinguished point of a space where the loops that define its fundamental group begin and end.

basis: A basis for a topological space X is a collection of open sets of X that contains "arbitrarily small" neighborhoods of every point of X. Specifically, for every point x of X, and for every open set U containing x, the collection must include a neighborhood of x lying within U.

bent: A bent path is a simple path having at most two segments.

blanket: A simply connected cover of a sheet.

border: A node of the adjacency graph borders the gates across which it has edges. It borders on a point $\alpha(0)$ in the direction of α if the first piece of the *partition* that α enters contains the region represented by that node.

borders: The borders of a piece P in a pattern for the sheet S are the components of $P \cap Bd\, S$. A border for the pattern is a border for any piece of the pattern.

boundary: I use this term only for manifolds. The boundary $Bd\, M$ of an manifold M is the set of points of M that have boundary patches. See Definition 2d.1. The boundary of an n-manifold is an $(n-1)$-manifold whose boundary is empty.

boundary patch: A *patch* $h: U \rightarrow H^n$ about a point $x \in U$ such that $h(x) \in R^{n-1} \subset H^n$.

boundary property: A set of potential cuts Ψ for a sketch S has the boundary property if all the configurations in $\mathbf{C}(S)$ that protect every potential cut of S lie within some closed subset of $\mathbf{C}(S)$.

bounding obstacle: In a sketch, a polygonal obstacle that encloses all the other features and the traces of the sketch. We add a bounding obstacle to a sketch for the purpose of relating it to designs.

branch: The branches of a design are the components of the inverse images of the design's articles (under the covering map). Two fringes in a blanket are in the same branch if and only if a link connecting them is degenerate. Branches are to blankets what articles are to sheets.

bridge: In a sketch, a path whose endpoints lie on features but whose middle intersects no feature. Traces are bridges, and the cuts of a sketch are the images of linear bridges.

bridge homotopy: A piecewise linear homotopy between bridges that moves their endpoints along their respective islands and moves their middles through the routing region. Two bridges are bridge-homotopic if there is a bridge homotopy that takes one to the other.

cable: In the RBE of a sketch, a group of rubber band segments (*strands*) with common endpoints.

CAD: Computer-aided design.

canonical: Also called 'parameterized by arc length'. A path α is canonical if the euclidean arc length of each subpath $\alpha_{s:t}$ is just $|t-s|$ times the arc length of α.

capacity: The capacity of a cut (or subcut) measures the amount of wiring space that the cut affords. It is defined as the arc length of the cut, measured in the *wiring norm*, decremented to account for the widths of the objects that contain the cut's endpoints.

chain: Any path in a manifold is a chain; it contains zero or more links. A chain for a path α is a chain that is path-homotopic to α.

channel: A simply connected *routing region*, usually rectangular. Also used informally to mean the routing space between two islands in a sketch.

chip: See *integrated circuit*.

clean: Making crossings only at its endpoints. A path in a sheet is clean in a design if it intersects the articles of the design at its endpoints alone.

closure: The closure of a subset A of a space X, denoted $Cl\,A$, is the minimal closed set of X that contains A.

coherent: Simple links in a blanket are coherent if they lift wires (or routes thereof) in the same design. See Definition 4c.2.

collapsible: Given a design Ω and an arrangement Γ, we say that a deviation $\omega_{s:t}$ of a wire in Ω across a subpath $\gamma_{a:b}$ of a cut in Γ is collapsible if $\omega_{s:t}$ is clean in Γ and $\gamma_{a:b}$ is clean in Ω.

compact: Compactness is a very important topological property. A topological space is compact if every collection of open sets that covers the space has a finite subset that also covers the space. The compact subspaces of R^n are the closed and bounded sets.

compaction: See *sketch compaction* and *layout compaction*.

component: Also called 'connected component'. The components of a topological space are its maximal connected subspaces. Two points of a space X lie in the same component of X if some connected subspace of X contains both points.

computational geometry: The study of algorithms that manipulate geometric objects.

concatenation: Formally, the concatenation of two paths α and β is the path $\gamma = \alpha \star \beta$ such that $\gamma_{0:t} = \alpha$ and $\gamma_{t:1} = \beta$, with $t = \frac{1}{2}$. Informally, we allow t to be any point in $(0,1)$.

condensed RBE: A form of the *rubber-band equivalent* in which the *strands* within each *cable* are not represented as separate entities. Instead, the condensed RBE records only the total with of the strands within each cable.

configuration: A vector of horizontal displacements of the *modules* in a *modular sketch*. The configuration $\mathbf{d} = (d_1, \ldots, d_n)$ for a modular sketch S corresponds to a sketch $S(\mathbf{d})$ in which module i has been shifted right by a distance d_i.

configuration space: The configuration space of a sketch is the set of configurations that preserve its topology. See Section 9A.

conform: A link ω conforms with an arrangement Γ if for every cut $\gamma \in \Gamma$, every crossing of γ by ω is necessary and no two are similar. A design Ω conforms with Γ if every link in Ω conforms with Γ.

congestion: The congestion of a cut in a layout measures the minimum amount of wiring that must cross the cut, regardless of how the wires (or traces) are routed. In most cases this quantity is equal to the *flow* across the cut.

connected: A topological space is connected if it cannot be partitioned into two disjoint, nonempty open sets. See *path-connected*.

constraint: In the context of compaction, an inequality relating the positions of two modules.

constraint graph: A edge-weighted, directed graph in which each vertex denotes a variable and each edge denotes a *simple linear inequality* between two variables. If x_k is the variable represented by vertex k, an edge of weight a_{ij} from vertex i to vertex j represents the constraint $x_j - x_i \geq a_{ij}$. By computing longest paths in the constraint graph, one can assign values to the variables so as to satisfy the constraints.

content: The sequence of wires, traces, or rubber bands that a cut necessarily crosses.

contractible: A space is contractible if it can be shrunk to a point within itself. The homotopy that does this is called a 'contraction'. Contractible spaces are *simply connected*.

convex: A subset X of R^n is convex if for every pair of points in X, the line segment between them also lies in X. A function $f: X \rightarrow R^1$ is convex if X is a convex subset of R^n for some n, and for every two points $x, y \in X$ and every point $t \in [0, 1]$, we have

$$f(tx + (1-t)y) \leq tf(x) + (1-t)f(y).$$

convexity property: A set of potential cuts Ψ has the convexity property if for each $\psi \in \Psi$, the capacity of $\psi(\mathbf{d})$ is a convex function of the configuration $\mathbf{d}$.

convolution: The convolution of two planar regions is the set of all vector sums of a point one region with a point in the other.

corridor· A sequence of line segments, called *doorways*, through which a path must pass; the input to Algorithm W. See Section 1B.

covering map: This one is hard to explain. See Definition 2b.1 and Figures 2b-1 and 2b-2.

covering space: The domain of a *covering map*; also called 'cover'. A space that looks locally like the space it covers, but whose parts may be connected together differently.

covering transformation: Also called 'deck transformation'. A covering transformation is a homeomorphism of a covering space with itself that preserves the covering map. For any two liftings of an object, there is a covering transformation that carries one to the other, provided that the covering space is connected and locally path-connected.

critical: A critical cut in a sketch is one that begins at a feature endpoint and travels to the closest point (in the wiring norm) on another feature. The critical cuts of a sketch are *decisive* in the sense that their safety and emptiness determine the routability of the sketch.

A critical *potential cut* is a potential cut χ_{pQ}, where p is a feature endpoint and Q is a feature, such that $\chi_{pQ}(\mathbf{d})$ is a critical cut from $p(\mathbf{d})$ to $Q(\mathbf{d})$ whenever one exists. See Definition 9c.1. Algorithm C uses critical potential cuts to generate routability constraints for sketch compaction.

crossing: Informally, a place where two paths meet. Formally, a pair (s, t) such that the image of s under the first path equals the image of t under the second path. In the design model, we allow $s, t \in [0, 1]$, while in the sketch model, we require $s, t \in (0, 1)$.

crossing sequence: In the sketch model, the crossing sequence of a ray $\overrightarrow{pq}$ at p is the sequence of rubber bands that cross over that ray at p, as defined in Section 1B. For the meaning of crossing sequence in the design model, see *plan*.

cross over: Two simple links in a blanket cross over if the image of one contains points in both *scraps* of the other.

curvilinear: Said of wiring norms: not piecewise linear.

cut: A path (or its image), often linear, used to test whether a layout is routable. The most important property of a cut is its *safety* or lack thereof.

cut: A simple link in a blanket cuts another link in the blanket if the terminals of the second link lie wholly one opposite sides of the first. See Definition 4b.1.

decisive: "Deciding routability". A set of cuts in a *sketch* is decisive if a sketch with the same set of features is routable if and only if all those cuts are either empty or safe in that sketch. By the sketch routability theorem, the critical cuts form a decisive set. A similar notion called '$\sharp$-decisiveness' is defined for sheets; see Definition 6d.1.

deformation retract: A subspace A of a space X is a deformation retract if X can be shrunk down to A without moving any point of A. The homotopy that does the shrinking is called a 'deformation retraction'.

degenerate: A cut is degenerate in a design if it is path-homotopic to a path in a single *article* of the design. This definition applies also to half-cuts and mid-cuts for wires in the design. More generally, a half-cut or mid-cut is degenerate if one of its *associated cuts* is degenerate. Degeneracy of cuts in sketches is similar, and is defined in Section 10C.

design: The more mathematical of my two basic representations of a circuit layer. Designs are defined in Section 4A. See also *sketch*.

design rules: Guidelines for the design of integrated circuits, intended to prevent unwanted behavior in the fabricated devices. For example, the design rules mandate a minimum separation between wires on the same layer, lest inaccuracies in the fabrication process cause the wires to short together.

details: The details of a design are its wires and fringes.

detour: A detour of a link around a barrier is a link that does not intersect the barrier, and that is formed by splicing in pieces of barrier's frontier into the original link. See Definition 5a.5. To find an evasive route of a wire in a safe sketch, we lift it and make detours around the barriers for the lifting.

deviation: A subpath $\omega_{s:t}$ of a wire is a deviation across a subpath $\gamma_{a:b}$ of a cut if $\omega_{s:t} \simeq_P \gamma_{a:b}$ or $\omega_{t:s} \star \gamma_{a:b}$ is a trivial link.

diagonal: Diagonal cuts are those that most strongly constrain the traces of a sketch; they have minimal capacity for cuts of their euclidean length. Formally, a cut is diagonal if its slope is *diagonal* and one of its endpoints is the vertex of a feature or fringe.

diagonal angles: The angles that correspond to the *diagonal slopes*.

diagonal slope: The wiring norm $\|\cdot\|$ defines the diagonal slopes: the slope of a line in R^2 is diagonal if for every two points p and q on the line, q is a vertex of the polygon $\{ x : \|x - p\| = \|q - p\| \}$.

discrete: A topological space X is discrete if every point of X is open in X. For example, the integers form a discrete subspace of the real line.

divide: A planar region $X \subset R^2$ divides a sheet if two fringes of that sheet fall in different components of $R^2 - X$.

divisive: An article of a design is divisive if its extent divides the design's sheet. Divisive articles are undesirable, for they may represent unwanted loops in the layout.

dominant: A set of cuts in a sheet is dominant if it dominates the set of all nontrivial straight cuts in that sheet. Dominant cut set are $\sharp$-decisive, by Corollary 6d.4.

dominate: One set of cuts in a sheet dominates another if every cut in the second set is *either weak* or can be reduced to a cut in the first set by a homotopy that does not increase its length. See Definition 6d.2. We exploit the relation of dominance to find small $\sharp$-decisive cut sets.

doorway: In a safe sketch, each necessary crossing of a cut by a trace has a nonempty doorway. The doorway is the portion of the cut where a feasible realization of the trace may locate that crossing.

dual graph: The dual of an embedded planar multigraph is the graph whose nodes are the faces of that graph, and which has an arc between two faces for each edge of the original graph that borders on those faces.

ECE: See *elastic-chain equivalent.*

edging: An edging for a sheet S is a finite set of convex polygons and line segments in $R^2 - (S - Bd\,S)$ whose union contains $Bd\,S$. See Definition 6d.7.

elastic: A canonical path is elastic if it has minimum euclidean arc length among all paths in its path-homotopy class.

elastic-chain equivalent: A set of chains obtained from a design by replacing each wire in the design by the elastic chain for some route of the wire. In the "standard" ECE, one replaces each wire by its own elastic chain.

element: An element of a sketch is a feature or trace in the sketch.

embedded: An embedded planar graph is one that comes with a specific drawing in the plane.

embedding: A map that is a homeomorphism onto its image. Also refers to a wire that is link-homotopic to a given wire, or a design that results from "re-embedding" (routing) the wires in another design.

empty: A cut is empty if its flow is zero and its endpoints lie on the same fringe or island. Even if a empty cut is unsafe, we can ignore it.

enclose: A loop in the plane encloses a set if it cannot be shrunk to a point without touching that set.

entanglement: The entanglement of a wire (or trace) with a cut is the minimum number of crossings of the cut by any route for the wire (or trace). It counts the crossings that cannot be "untangled" by routing the wire. Compare *winding.*

equivalent: Two covering spaces of the same base space are equivalent if they are homeomorphic in a way that leaves the covering maps unchanged. See Proposition 2b.7. Two configurations of a modular sketch are equivalent with respect to a potential cut ψ if in moving linearly from one to the other, ψ is always a cut.

essential: Not path-homotopic to a constant loop. This definition is not entirely consistent with standard terminology, which defines 'essential' as "not homotopic to a constant map". I do not need the latter concept, however.

euclidean: The euclidean norm $|\cdot|$ is defined by $|(x, y)| = \sqrt{x^2 + y^2}$.

evasive: Avoiding its *barriers.* In the design model, a route of a wire is evasive if it has no unsafe, straight, nontrivial half-cuts.

eventually: Section 8A defines for each suitably restricted sketch a family of sheets and designs parameterized by a positive real number ϵ. A statement involving ϵ holds eventually if it holds for all ϵ less than some $\epsilon_0 > 0$.

exposed: A cut α in a sketch is exposed if the corresponding cut α^b eventually satisfies $\|\alpha^b\| = \|\alpha\| - 2\epsilon$.

extent: Essentially a synonym for *territory.* The details of a design have extents, whereas the elements of a sketch have territories. Anyway, the extent of a detail of *width d* is the set of points closer than $d/2$ units to that detail, as measured in the *wiring norm.*

face: The faces of an embedded planar graph are the regions into which the edges of that graph divide the plane. The "outer" face is the unique unbounded one.

feasible: In general, a realization of a wire in a routing problem is feasible if it is part of a correct solution to the routing problem. Thus, an embedding of a wire in a design is feasible if some proper embedding of the design contains it.

feature: An inflexible object in a sketch. Every feature is a point or line segment.

flat: A n-manifold is flat if it comes with a local embedding into R^n. Flat manifolds include sheets, blankets, and scraps of blankets.

flow: The flow across a cut is a weighted sum of the *necessary* crossings of that cut by wires, where each crossing is weighted by the width of its wire. (Actually, flow counts equivalence classes of necessary crossings, rather than the crossings themselves.) Flow and congestion are equal for simple cuts, but flow is the deeper and more important of the two concepts. The notion of flow makes sense in all the routing problems I consider, although I define it formally only for the design model.

forbidden: Said of half-links in a blanket: contributing to a *barrier*. A half-link σ is forbidden to a wire lifting $\tilde{\omega}$ if to route $\tilde{\omega}$ through $\sigma(1)$ would keep ω from being *evasive*. See Definition 5a.1.

forbidden zone: The union of the left-hand or right-hand barriers for a wire lifting.

free: A path is free in a pattern if no seam in the pattern contains either endpoint of the path.

fringe: A component of the boundary of an n-manifold. A fringe is a path-connected $(n-1)$-manifold, closed in its parent manifold. The fringes of a sheet form the terminals and routing obstacles of the designs on that sheet.

frontier: The frontier of a subset A in a space X, denoted $Fr\,A$, is $Cl\,A - Int\,A$: the set of points in the closure of A not in the interior of A.

full plan: The full plan of α in an arrangement Γ is the plan containing all the crossings of the cuts in Γ by α, sorted by position along α. It makes sense only when the crossings of α in Γ are discrete.

fundamental group: An algebraic structure on the path classes of loops in a space at a given *base point*. See Definition 2a.3. The fundamental group of a space is an important topological invariant, part of the study of algebraic topology.

gap: A portion of a chain between two major links of the chain.

gate: A straight path forming part of a *tunnel* or a *partition* of a sketch.

gate arc: In the adjacency graph of a sketch, an arc representing adjacency across a gate.

gate list: The sequence of gates that a path crosses over, whether in the routing region of a sketch or in its *adjacency graph*.

graph: A mathematical structure comprising a set of "vertices", also called 'nodes', and a set of "edges", also called 'arcs', each of which is "incident" on exactly one or two vertices. Often the edges and vertices have additional information attached to them.

grid: The set of points in the plane which have at least one integral coordinate. The lines in this set are called 'gridlines', and the points where these lines intersect are called 'gridpoints'.

grid-based: Refers to a *wiring model* in which wires are constrained to run in a grid of horizontal and vertical lines.

half-cut: A half-link between a fringe and a route of a wire, used to measure the flow between the fringe and the wire.

half-link: A path α in a manifold that touches the manifold's boundary at $\alpha(0)$ only.

half-thread: The image of a simple half-link.

Hausdorff: In a Hausdorff space, every two distinct points have disjoint neighborhoods. All the spaces I consider are Hausdorff.

height: The height of a potential cut ϕ_{pq}, whose endpoints do not move vertically, is the difference between the y-coordinates of p and q.

homeomorphism: A continuous, bijective function with a continuous inverse.

homotopy: A 'continuous deformation' or 'continuous family' of topological maps. See Definitions 2a.1 and 2a.6.

IC: See *integrated circuit.*

ideal: An ideal route of a wire is canonical, evasive, and as short as possible. Ideal embeddings are wires, and form a design; anything associated with this design is also called ideal. We route every wire in a safe design by means of its ideal embedding. By analogy with designs, we also apply the term 'ideal' to sketches; every trace in a routable sketch has an ideal realization, and these form a proper realization of the sketch.

inner: Said of fringes in a sheet: an inner fringe is one whose inside is not part of the sheet. Every sheet has at least one inner fringe. Compare *outer.*

inside: Every simple loop in a blanket or in the plane has an 'inside' and an 'outside'. The inside of a blanket loop includes no part of any fringe.

integrated circuit: An electronic device made by depositing materials in and on a wafer of semiconducting material in precisely controlled patterns. Often called 'chips' or (in *the popular press*) '*microchips*', integrated circuits are the computational elements at the heart of every modern digital computer.

interior: The interior of a subset A of a space X, denoted $Int\,A$, is the maximal open set of X contained in A.

intersection graph: The intersection graph of a sketch and a *partition* of that sketch is the graph whose nodes are features and the line segments where traces and gates intersect, and whose arcs are the subpaths of features and traces that connect these regions.

island: A maximal connected group of features in a sketch.

jog: A *joint* of a wire or trace.

jog point: A point at which at which a wire is allowed to develop a jog during compaction.

joint: A point (in the unit interval I) at which a piecewise linear path is not linear.

kinship: See *akin.*

layout: In general, the geometric structure of a circuit design. I use the term 'layout' to refer to an instance of a wire-routing problem, such as a sketch or design.

layout compaction: In general, the problem of minimizing the area of a circuit layout by altering its geometry.

leaf cell: The simplest modules in a VLSI design aside from transistors and other basic devices.

lift: Also called 'lifting'. In the context of a covering map $p: M \rightarrow X$, a lift of a map $g: C \rightarrow X$ is any map $\tilde{g}: C \rightarrow M$ such that $p \circ \tilde{g} = g$. Outside of Chapter 2, the covering map p is always taken to be the covering of a sheet by its blanket.

lifting: The process of converting maps into a base space into maps into its covering space.

line segment: A line segment is the image of a straight path.

linear programming: A classical and very important optimization problem: maximize a given linear function of real-valued variables subject to specified linear inequalities. Linear programming is solvable in polynomial time.

link: A path in a manifold that touches the manifold's boundary at its endpoints alone.

link code: The sequence of cuts in an *arrangement* necessarily crossed by a link or a chain for a link. See Definition 7b.1.

link homotopy: A homotopy between links that moves their endpoints along their respective fringes; or the relation of being link-homotopic. Two links are link-homotopic if there is a link homotopy (in the first sense) between them.

link plan: A sequence of crossings that a link (or a chain for a link) is forced to make with cuts in an arrangement, given that its link class is fixed. See Definition 7b.1.

list: A sequence of paths that a given path crosses over. See, for example, *seam list*.

local: A property of topological spaces is usually said to hold *locally* in a space X if it holds within arbitrarily small neighborhoods of every point of X. (For properties that open sets do not normally have, such as compactness, the definition has to be modified somewhat.) For example, a space is locally *path-connected* if it has a basis of path-connected sets.

local embedding: The map $f: X \rightarrow Y$ is a local embedding if X has a basis of open sets U such that $f|_U$ is an embedding.

local homeomorphism: The map $f: X \rightarrow Y$ is a local homeomorphism if X has a basis of open sets U such that $f(U)$ is open in Y and $f|_U$ is an embedding.

locally minimal: A linear path between two fringes of a sheet is locally minimal if its length (in the wiring norm) cannot be reduced by moving its endpoints along their respective fringe edges. The path need not be a chain; it can leave the sheet.

loop: A path whose endpoints coincide. A loop of k links is...

major: Neither empty nor degenerate (said of cuts and links).

manifold: A topological space that is locally homeomorphic to R^m for some m. See Definition 2d.1.

margin: The margin of a cut is the difference between its capacity and its flow. Safe cuts are those with nonnegative margin (of safety). A subcut whose margin is zero is called 'marginal', or 'marginally safe'.

maze: A collection of *tunnels*, indexed by pairs $\pm\delta$ of *diagonal angles*, which begin and end at the same points. Every gate in the tunnel corresponding to the angles $\pm\delta$ must be a subpath of a linear path of angle $\pm\delta$.

metric: Also called 'distance metric', a metric on a set P is a function d from $P \times P$ to the nonnegative real numbers, satisfying three axioms: (1) $d(p,q) = 0$ if and only if $p = q$; (2) $d(p,q) = d(q,p)$ for all $p,q \in P$; and (3) the triangle inequality, $d(p,q) + d(q,r) \geq d(p,r)$ for all $p,q,r \in P$. The metric d gives rise to a topology on P; a basis for this topology is the collection of sets $\{ q : d(p,q) < \epsilon \}$ for $p \in P$ and $\epsilon > 0$. In other words, a subset S of P is open in the metric topology if for every point $p \in S$ there is a number $\epsilon > 0$ such that S contains the set $\{ q : d(p,q) < \epsilon \}$.

metric space: A topological space whose topology is given by a metric. All the spaces considered in this thesis are metric spaces.

mid-cut: A mid-link between two routes of wires, or between two points on the same route.

mid-link: A path in a manifold that does not intersect the manifold's boundary.

middle: The middle of a path α is the set $\alpha\big((0,1)\big)$.

minimal: A minimal path from a compact region P to a compact region Q is a linear path from P to Q whose arc length, measured in the wiring norm, is the distance $\|P - Q\|$ from P to Q.

minor: Either empty or degenerate (said of cuts and links).

modular: A modular sketch is a sketch together with a grouping of its islands into modules.

module: A set of sketch islands that move as a unit during compaction.

multigraph: A graph in which each pair of nodes can have multiple arcs between them.

multiply connected: Not *simply connected.*

necessary: Informally, a crossing of a cut by a wire (or trace, bridge, or link) is necessary if it cannot be removed by applying a homotopy (of the appropriate type) to the wire. The design model provides a formal definition (4b.2).

neighborhood: A neighborhood of a point or set is an open set that contains it.

net: A set of terminals to be connected, or a wire that connects them. Usually appears as 'multiterminal net', to contrast with the usual two-terminal nets. In the sketch model a net is a loop of traces that do not cross over and enclose no features. In the *network model* a net is a loop whose terminals are :he islands it encloses.

network: A collection of nonintersecting nets and islands; an instance of a proposed wiring model (see Section 10D).

norm: A norm provides a uniform way of measuring distances in a vector space. A map $\|\cdot\|$ from a vector space to the nonnegative real numbers is a norm if three conditions hold: (1) $\|x\| = 0$ if and only if x is the zero vector; (2) $\|tx\| = |t| \cdot \|x\|$ for all vectors x and real numbers t; and (3) $\|x + y\| \leq \|x\| + \|y\|$ for all vectors x and y. The distance between x and y in the norm $\|\cdot\|$ is just $\|x - y\|$. See also *wiring norm.*

normal: Said of topological spaces. In a normal space, every two disjoint closed sets have disjoint neighborhoods. All metric spaces are normal.

obstacle: An *island* of a sketch that is not a *terminal.*

ordering property: A property required of the sequence of potential cuts input to Algorithm A, my abstract compaction algorithm. See Section 9E.

outer: Said of a fringe in a sheet. Every sheet has exactly one outer fringe, within which the rest of the sheet lies.

outside: The outside of a simple loop in a space X consists of every point in X that is neither on the loop nor *inside* it.

partial realization: A partial realization of a trace is minimum-length path through the gates for that trace of a particular diagonal slope. Partial realizations are constructed and used by Algorithm T.

partial route: A partial route for a maze is a minimum-length path through one of the tunnels of the maze.

partition: A partition of a sketch is a set of straight, horizontal cuts in the sketch that slice each component of its routing region into simply connected pieces.

patch: A patch about a point x in an n-*manifold* is a *homeomorphism* of a *neighborhood* of x with an open set in the half-space H^n.

path: A continuous function with domain $I = [0, 1]$. See the beginning of Chapter 2 for definitions related to paths.

path class: An equivalence class under the relation of path homotopy.

path code: In general, the sequence of cuts in an *arrangement* necessarily crossed by a path. See Definition 7b.1. When the arrangement is a pattern, the path can be constructed by reducing the *seam list* of the path in that pattern.

path component: The path components of a topological space are its maximal path-connected subsets. Two points lie in the same path component of a space X if there is a path in X from one to the other.

path-connected: A topological space is path-connected if every pair of its points can be connected by a path. Every path-connected space is connected, but not vice versa. If a space is connected and *locally* path-connected, however, then it is path-connected.

path homotopy: A homotopy between paths that fixes their endpoints; or the relation of being path-homotopic. See Definition 2a.1. Two paths are path-homotopic if there is a path homotopy between them.

path plan: A sequence of crossings that a path is forced to make with cuts in an arrangement, given that its path class is fixed. See Definition 7b.1.

pattern: A set of straight cuts called *seams* that divide a sheet into simply connected *pieces* for the purpose of determining which paths are path-homotopic. See Definition 7a.1.

PCB: See *printed circuit board.*

piecewise: In general, a property holds piecewise for a map $f: X \rightarrow Y$ if X can be "triangulated" (divided into simplices) such that f has this property when restricted to each simplex. See the next entry.

piecewise linear: A map $f: X \to Y$ is piecewise linear if $X \subset R^n$ for some n, and X can be chopped into simplices (points, line segments, triangles, tetrahedra, etc.) such that f is linear on each simplex, and only finitely many simplices meet at each point. The composition of piecewise linear maps is piecewise linear, and the inverse of a piecewise linear map is piecewise linear.

pivotal: The pivotal cuts in a sketch are the diagonal cuts and the cuts between feature endpoints. Like the *critical* cuts, they are decisive.

PL: An abbreviation for *piecewise linear.*

placement problem: A problem that involves positioning inflexible objects (modules) as well as flexible ones (wires).

plan: A plan for a path ω is a finite sequence of triples (γ, a, t) such that $\gamma(a) = \omega(t)$. Usually the paths γ are taken from some arrangement Γ. See also *full plan*, *path plan*, and *link plan*.

planar: A graph is planar if its vertices and edges can be drawn in the plane without crossovers.

pointlike: A pointlike feature in a sketch is one that intersects no other features in the sketch and consists of a single point.

polygonal: A subset of the plane is polygonal if it lies within the union of a polygon with its inside, and contains the inside of the polygon. A wiring norm is polygonal if the set of points of norm 1 is a polygon.

potential cut: A linear path between two features of a sketch that moves in a continuous manner as those features move, depending only on their relative position. In any particular configuration, a potential cut may or may not give rise to a cut; hence the name. See Section 9C.

printed circuit board: A support and connector for electronic devices, made by plating metal wires onto layers of insulating material.

proper: Representing a valid circuit layout: "design-rule correct". A sketch is proper if its traces are *self-avoiding*, and whenever two elements of the sketch have overlapping *territories*, they are a trace and one of its terminals. The corresponding property of designs is denoted by the term '$\natural$-proper'. A design is proper if its wires are self-avoiding and its articles have disjoint extents.

protect: A *configuration* $\mathbf{d}$ protects a *potential cut* ψ for the sketch S if in the sketch $S(\mathbf{d})$, the path $\psi(\mathbf{d})$ is either a safe cut or not a cut at all.

quotient space: A space obtained from another by identifying or "gluing" some points to some others. Formally, Y is a quotient space of X if there is a surjective map $f: X \to Y$ such that the open sets of Y are those sets $U \subseteq Y$ for which $f^{-1}(U)$ is open in X.

rail: A rail of a track ω is a segment of ω that is either (1) supported at only one end, or (2) supported at both ends by ties of the same slope.

RBE: See *rubber-band equivalent.*

reachable: One sketch is reachable from another if it can be obtained from the other sketch by a continuous motion of modules and wires that shifts modules horizontally and maintains the routability of the sketch.

realization: A trace or sketch that is the result of a routing process; it may or may not be *feasible*.

rectilinear: Composed of horizontal and vertical segments. The rectilinear norm on R^2 is defined by $\|(x,y)\| = \max\{|x|,|y|\}$.

reduced seam list: See *path code*.

reduced intersection graph:

reflect: Two paths in a blanket reflect a *crossing* between their projections if they make that crossing themselves.

region: Usually refers to a subset of the plane.

respect: A relation that may obtain between a cut (or subcut) and a design; see Definition 4e.1. Respect and weak respect (Definition 4c.6) are the main technical conditions that permit us to relate the flows of different subcuts. A half-cut or mid-cut respects a design (strongly or weakly) if all its associated cuts respect the design (strongly or weakly).

restrain: A sheet S (or a gate γ) restrains a path α at x if for all sufficiently small open intervals (s,t) containing x, the path $\alpha(s) \triangleright \alpha(t)$ leaves S (or fails to intersect $Im\,\gamma$).

restricted route: An alternate definition of *partial route*; see Section 7E.

retract: A subspace A of a space X is a retract if there is a map $f: X \to A$ that fixes every point of A. The map f is called a 'retraction'.

rigid: Straight, nondegenerate, and marginal (a property of subcuts).

river routing: Refers to wire-routing problems in which wires do not change layers. Thus all single-layer routing problems may be considered river routing problems, but I prefer to reserve the term 'river routing' for situations in which each component or layer of the routing region is *simply connected*.

roots: With respect to a pattern in which α is free, the roots of a path α are the borders of that pattern that contain the endpoints of α.

rough routing: A path that indicates the *path class* of a wire to be routed.

routable: An instance of a routing problem (e.g., a sketch or design) is routable if it has a proper routing (realization, embedding). Similarly, a design is $\sharp$-routable if it has a $\sharp$-proper embedding.

routability conditions: Necessary and sufficient conditions for a layout to be routable.

routability property: A set Ψ of potential cuts for a sketch S has the routability property if (1) the failure of a configuration $\mathbf{d}$ to protect all elements of Ψ implies unroutability of $S(\mathbf{d})$, and (2) the routability of $S(\mathbf{d})$ is guaranteed if all configurations $t\mathbf{d}$ with $t \in [0,1]$ protect all elements of Ψ.

route: A route for a trace is any bridge, not necessarily a trace, that is bridge-homotopic to that trace. A route for a wire is any link, not necessarily a wire, that is link-homotopic to that wire.

route: Also refers to a path through a *tunnel*. If ω is a tight track through a maze and δ is a diagonal angle, then the shortest path through the δ-tunnel of this maze is called the δ-route of ω.

routing region: In an instance of a routing problem, the space through which the wires are to be routed.

rubber band: The rubber band for a trace in a sketch is the shortest path that is a limit of routes for that trace.

rubber-band equivalent: A standard form for a sketch; the input to my routing and routability testing algorithms. The rubber-band equivalent (RBE) represents the features of the sketch and the *rubber bands* that result from shrinking each trace to its minimum length. The RBE data structure is optimized for computing the sequences of necessary crossings of cuts in the sketch.

safety: The central concept in the routability theorems concerning single-layer routing. A cut is safe if and only if its congestion (or flow) does not exceed its capacity. (Where flow and congestion are both defined, we use flow to determine safety.) A sketch (or design) is safe (or $\sharp$-safe) if and only if all its nonempty straight cuts are safe. A design is safe if and only if all its major straight cuts are safe.

scanning: A fundamental algorithmic technique in computational geometry. A scanning algorithm constructs its output by sweeping a *scan line* across the objects in its input, processing each object as it enters and leaves the scan line.

scrap: A simply connected, open submanifold of a *blanket*.

seam: One of the straight cuts in a *pattern*.

seam list: The sequence of seams in a pattern that a piecewise linear path croses over.

segment: The segments of a piecewise linear path α are its maximal linear subpaths $\alpha_{s:t}$ with $s < t$. Consecutive segments of a PL path can be collinear if the path is not canonical.

self-avoiding: A wire in a design is self-avoiding if its article does not *divide* the sheet. Similarly, a trace in a sketch is self-avoiding if its territory, together with those of its terminals, does not separate any two of the sketch's islands. The requirement that wires be self-avoiding is one of the complications of wire routing in multiply connected regions.

semisimple: Semisimplicity is a desirable attribute of half-cuts and mid-cuts. The subpaths of a cut between its necessary crossings by wires are semisimple subcuts for those wires. All subcuts *akin* to these are semisimple as well. See Definition 4e.5.

separable: A separable space is one that has a countable dense subset.

settle: Section 8A defines for each suitably restricted sketch a family of sheets and designs parameterized by a positive real number ϵ. A function f of ϵ settles at a function g of ϵ if the equality $f(\epsilon) = g(\epsilon)$ holds holds for all ϵ less than some $\epsilon_0 > 0$.

shadow: The shadow cast by a point $r \in R^2$ with respect to a point p is the set of points q such that $||p - q|| = ||p - r|| + ||q - r||$.

shadowed: A cut $\overline{pq}$ in a sketch is shadowed if there is a point r on a feature of the sketch such that q is in the shadow of r with respect to p.

sheet: The routing region for a *design*; the result of removing one or more (but finitely many) polygonal holes from a closed polygonal region in the plane.

side: A simple link in a blanket separates it into two scraps, one on its left and one on its right. These scraps are the two sides of the link.

similar: Two crossings between paths in a sheet are similar if the liftings that reflect one also reflect the other. Equivalently, the crossings are similar if the subpaths that connect them are path-homotopic. See Definition 4b.2.

simple linear inequality: In the context of linear programming, an inequality $x_j - x_i \geq a_{ij}$ in which x_i and x_j are variables and a_{ij} is a constant.

simple loop: A piecewise linear loop that would be injective but for the coincidence of its endpoints.

simple path: A piecewise linear and injective path.

simply connected: A topological space is simply connected if (1) it is path-connected. and (2) every loop in that space can be continuously shrunk to a point. For a formal definition, see Definition 2a.4.

skeleton: The subgraph of an adjacency graph obtained by omitting gate arcs.

sketch: One of my two basic representations of a circuit layer, discussed in Section 1A.

sketch compaction: Given a routable sketch, the problem of finding and routing a *reachable* sketch of minimum width. See Section 9A.

sketch routability: The problem of determining whether a given sketch is routable.

sketch routing: Given a routable sketch, the problem of finding a proper realization that minimizes the euclidean arc length of every trace.

space: A topological space: a set with a system of neighborhoods (open sets) closed under finite intersection and arbitrary union.

span: A cut set Γ **spans** the sheet S if for some edging Δ of S and for every two elements $P, Q \in \Delta$ such that the minimal cuts from P to Q are all cuts in S, the set Γ a minimal path from P to Q.

stable: A design is stable with respect to an arrangement if wherever a wire in the deisgn intersects a cut in the arrangement, it intersects transversely, crossing over the cut at that point.

starlike: Also called 'star-convex'. A subset P of a flat manifold is starlike about $x \in P$ if for every point $y \in P$, the linear path $x \triangleright y$ exists and lies in P. A convex set is one that is starlike about each of its points.

straight: A path in a flat m-manifold is straight if its projection to R^m is linear and nonconstant.

strand: One *segment* of a rubber band. (Rubber bands are piecewise linear.)

string: A finite sequence over a fixed alphabet. Path codes and link codes are strings over a pattern.

strut: A *rigid* cut or half-cut around which a wire route bends. See Definition 5b.5 and Section 1D. The struts of an ideal embedding are the constraints that force it to be as long as it is.

subcut: A cut, half-cut, or mid-cut.

sublink: A subpath of a link; any path in a manifold whose middle does not intersect the manifold's boundary.

submanifold: A subset of a manifold that is itself a manifold of the same dimension.

subpath: A subpath of a path α is any path of the form $\alpha_{s:t}$ for $s, t \in I$. The definition of $\alpha_{s:t}$ is $\alpha_{s:t}(x) = \alpha\big((1-x)s + xt\big)$.

A *track* has certain special subpaths, called δ-subpaths, for each diagonal angle δ. A δ-subpath of a track ω is a path $\omega_{s:t}$ with $s < t$ such that either $s = 0$ or ω has a tie of angle $\pm\delta$ at s, and either $t = 1$ or ω has a tie of angle $\pm\delta$ at t.

subspace: A subset A of a topological space X with the inherited topology: the open sets in A are the intersections of the open sets of X with A.

substring: A contiguous subsequence of a string.

support: A straight path σ in R^2 supports a piecewise linear path ω at $s \in (0,1)$ if $\sigma(1) = \omega(s)$ and ω turns toward $\sigma(0)$ at s. If $\omega_{r:s}$ and $\omega_{s:t}$ are segments of ω, we also say that σ supports these segments.

tangent: A straight path α in R^2 is tangent to a straight path σ if the line containing α intersects the polygon $P(\sigma)$ at $\sigma(1)$, but does not intersect $inside(P(\sigma))$.

taut: A route of a wire is taut if it has a strut at each of its joints. Ideal routes are taut (Proposition 5b.6).

taxicab: The taxicab norm on R^2 is defined by $\|(x,y)\| = |x| + |y|$.

terminal: In general, the terminals of a wire (or trace) are the fixed objects to which that wire must connect. The terminals of a link or half-link are the fringes that contain its endpoints.

terminal merging: Refers to a modification of the sketch model in which the terminals of each trace are permitted to have overlapping territories and to coalesce during compaction.

territory: The territory of an object (feature, trace, fringe, or wire) is a region of the plane that represents the space allocated to it on its layer. It accounts not only for the physical dimensions of the object, but also for the necessary separation between objects. In other words, it encapsulates the geometric design rules for that object; two objects are assumed to interact if and only if their territories overlap.

thread: The image of a simple link.

tie: A tie for a *track* ω is a straight path σ whose angle is diagonal, and which supports two segment of ω, both of which are tangent to σ.

tight: A piecewise linear path α is tight in a sheet S if S *restrains* ω at each of its joints. Similarly, α is tight in a tunnel or maze if for each joint x of α, some gate in that tunnel or maze restrains α at x.

topological property: A property that is preserved by homeomorphisms; what topology is about.

trace: A flexible object in a sketch. The metallized lines on a printed circuit board are (or used to be) called 'traces'.

trace arc: In the adjacency graph of a sketch, an arc representing adjacency across a gate.

trace homotopy: A *bridge homotopy* that fixes the endpoints of the bridge.

track: A piecewise straight path with a *tie* at every joint.

trivial: Of paths in sheets, path-homotopic to a path in a single fringe. In a sheet S, a crossing (c, r) of a cut χ by a chain ρ is trivial if for some $i, j \in \{0, 1\}$ the path $\chi_{i:c} \star \rho_{r:j}$ homotopic to a path in $Bd\, S$.

tubular neighborhood: An especially nice neighborhood of a simple sublink; see Definition 3b.3.

tunnel: A sequence of *gates* through which a path must pass. Through any tunnel there is a unique minimum-length path. Tunnels are similar to corridors, but are more precisely defined; see Definition 7d.5.

turning: A piecewise linear path α turns at $s \in (0, 1)$ if α has two segments $\alpha_{r:s}$ and $\alpha_{s:t}$ which either overlap or form an angle. If α is a link in a sheet, then α turns at $x \in \{0, 1\}$ if it forms an acute angle with a fringe there. If α is a path in the plane, it turns 'toward' some points and 'away from' others. If α is a link in a blanket, it turns 'toward' one of its scraps and 'away from' the other.

uniform convergence: A sequence of functions $\langle f_n \rangle$ into a *metric space* with metric d converges uniformly to a function f if for every $\epsilon > 0$ there is an N such that $d(f(x), f_n(x)) < \epsilon$ for all $n \geq N$ and all x.

unit polygon: For a piecewise linear norm, the analogue of the unit circle: the set of vectors of norm 1.

unsafe: See *safety*.

via: A connection between wires on different layers of a chip or printed circuit board. In an integrated circuit, also called 'contact cut'.

visibility graph: The visibility graph of a sketch is a function of the features in the sketch. Its nodes are the feature endpoints and its arcs are the features and the cuts between feature endpoints.

VLSI: Stands for Very-Large-Scale Integration; refers to the technology that allows millions of electrical devices to be fabricated on a single *chip*.

wall: When compacting a sketch horizontally, we assume that the sketch is bounded at the left and right by vertical lines. These lines are called walls, and are treated as features.

weak: If a straight cut in a sheet can be reduced to a straight chain by link and path homotopies that do not increase its length, and this chain contains either two or more links or an entire fringe edge, then the cut is weak. See Definition 6d.2. One may ignore weak cuts when testing routability.

weak respect: See *respect* (and Definition 4c.6).

web: A web of k threads is the image of a *loop of k links* in a blanket.

width: Every feature and trace in a sketch, and every fringe and wire in a design, has a width that indicates how much area it requires. See *extent* and *territory*. In the RBE of a sketch, a crossing sequence or a cable has a width equal to the sum of the widths of the rubber bands it involves.

winding: The winding of a cut and a wire is the number of *similarity* classes of necessary crossings between them. Winding is to entanglement as flow is to congestion.

wire: Something to be routed. In the design model, 'wire' has a technical meaning: a wire in a sheet is a simple link whose terminals are convex and *inner*.

wiring model: The set of rules (*design rules* and others) that determine how the wires in a routing problem may be routed. More specifically, the definition of what constitutes a *proper* solution of a routing problem.

wiring norm: Part of a wiring model: the norm $\|\cdot\|$ used to measure widths, extents, and separations of layout components, and the capacities of cuts. Normally the wiring norm is an arbitrary *piecewise linear* norm, which means that the set of points of norm 1 is a convex polygon.

References

[1] B. S. Baker, S. N. Bhatt, and F. T. Leighton, "An approximation algorithm for Manhattan routing," in *Advances in Computing Research*, Vol. 2, editor F. P. Preparata (JAI Press, Greenwich, CT, 1984), pp. 205–229.

[2] B. S. Baker and R. Y. Pinter, "An algorithm for the optimal placement and routing of a circuit within a ring of pads," *24th Annual Symposium on Foundations of Computer Science* (November 1983), pp. 360-370.

[3] M. Becker and K. Mehlhorn, "Algorithms for routing in planar graphs," *Acta Informatica*, Vol. 23, No. 2 (April 1986), pp. 163–176.

[4] B. Berger, *New Upper Bounds for Two-Layer Channel Routing*, M.S. Thesis, MIT Department of Electrical Engineering and Computer Science (January 1986); MIT VLSI Memo No. 86-312.

[5] M. L. Brady and D. J. Brown, "Arbitrary planar routing with four layers," *Proceedings, Conference on Advanced Research in VLSI* (January 1984), pp. 194–201.

[6] R. Cole and A. Siegel, "River routing every which way, but loose," *25th Annual Symposium on Foundations of Computer Science* (October 1984), pp. 65–73.

[7] R. Cole and C. K. Yap, "Geometric retrieval problems," *24th Annual Symposium on Foundations of Computer Science* (November 1983), pp. 112–121.

[8] D. Dolev, K. Karplus, A. Siegel, A. Strong, and J. D. Ullman, "Optimal wiring between rectangles," *13th Annual ACM Symposium on Theory of Computing* (May 1981), pp. 312–317.

[9] M. Doreau, private communication (1983).

[10] A. E. Dunlop, "SLIP: symbolic layout of integrated circuits with compaction," *Computer Aided Design*, Vol. 10, No. 6 (November 1978), pp. 387–391.

[11] H. Edelsbrunner and L. J. Guibas, "Topologically sweeping an arrangement," *18th Annual ACM Symposium on Theory of Computing* (May 1986), pp. 389–403.

[12] M. L. Fredman and R. E. Tarjan, "Fibonacci heaps and their uses in improved network optimization algorithms," *25th Annual Symposium on Foundations of Computer Science* (October 1984), pp. 338–346.

[13] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, New York (1979).

[14] S. Ghosh and D. Mount, "An output sensitive algorithm for computing visibility graphs," *28th Annual Symposium on Foundations of Computer Science* (October 1987), to appear.

[15] L. Guibas, private communication (March 1987).

[16] M. Y. Hsueh, *Symbolic Layout and Compaction of Integrated Circuits*, Ph.D. thesis, EECS Division, University of California, Berkeley, CA (1979).

[17] M. Kaufmann and K. Mehlhorn, "Routing through a generalized switchbox," *Journal of Algorithms*, Vol. 7, No. 4 (December 1986), pp. 510–531.

[18] G. Kedem and H. Watanabe, "Graph-optimization techniques for IC layout and compaction," *20th Design Automation Conference* (June 1983), pp. 113–120.

[19] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science*, Vol. 220, No. 4598 (May 1983), pp. 671–680.

[20] M. R. Kramer and J. van Leeuwen, "Wire-routing is NP-complete," Technical Report RUU-CS-82-4, Department of Computer Science, University of Utrecht, the Netherlands (February 1982).

[21] C. E. Leiserson and F. M. Maley, "Algorithms for routing and testing routability of planar VLSI layouts," *17th Annual ACM Symposium on Theory of Computing* (May 1985), pp. 69–78.

[22] C. E. Leiserson and R. Y. Pinter, "Optimal placement for river routing," *SIAM Journal on Computing*, Vol. 12, No. 3 (August 1983), pp. 447–462.

[23] C. E. Leiserson and J. B. Saxe, "A mixed-integer linear programming problem which is efficiently solvable," *21st Annual Allerton Conference on Communication, Control, and Computing* (October 1983), pp. 204–213.

[24] T. Lengauer, "Efficient algorithms for the constraint generation for integrated circuit layout compaction," *9th International Workshop on Graphtheoretic Concepts in Computer Science* (June 1983), pp. 219–230.

[25] T. Lengauer and K. Mehlhorn, "The HILL system: a design environment for the hierarchical specification, compaction, and simulation of integrated circuit layouts," *Proceedings, Conference on Advanced Research in VLSI* (January 1984), pp. 139–149.

[26] T. Lengauer, "On the solution of inequality systems relevant to IC layout," *Journal of Algorithms*, Vol. 5, No. 3 (September 1984), pp. 408–421.

[27] R. J. Lipton and R. E. Tarjan, "A separator theorem for planar graphs," *SIAM Journal on Applied Mathematics*, Vol. 36, No. 2 (April 1979), pp. 177–189.

[28] F. M. Maley, "Compaction with automatic jog introduction," *1985 Chapel Hill Conference on VLSI* (May 1985), pp. 261–283.

[29] F. M. Maley, *Compaction with Automatic Jog Introduction*, M.S. Thesis, MIT Department of Electrical Engineering and Computer Science (May 1986); Technical Report TR-372, MIT Laboratory for Computer Science.

[30] F. M. Maley, "An observation concerning constraint-based compaction," *Information Processing Letters*, Vol. 25, No. 2 (May 1987), pp. 119–122.

[31] C. Mead and L. Conway, *Introduction to VLSI Systems*, Addison-Wesley, Menlo Park, California (1980).

[32] K. Mehlhorn and F. P. Preparata, "Routing through a rectangle," *Journal of the ACM*, Vol. 33, No. 1 (January 1986), pp. 60–85.

[33] S. Gao, M. Jerrum, M. Kaufmann, K. Mehlhorn, W. Rülling, and C. Storb, "Homotopic one-layer routing", private communication (July 1987).

[34] A. Mirzaian, "Channel routing in VLSI," *16th Annual ACM Symposium on Theory of Computing* (May 1984), pp. 101–107.

[35] E. Moise, *Geometric Topology in Dimensions 2 and 3*, Springer-Verlag, New York (1977).

[36] R. C. Mosteller, *Monte Carlo Methods for 2-D Compaction*, Ph.D. Thesis, California Institute of Technology (1986).

[37] R. C. Mosteller, A. H. Frey, and R. Suaya, "2-D compaction, a Monte Carlo method," *Advanced Research on VLSI: 1987 Stanford Conference* (April 1987), MIT Press, pp. 173–197.

[38] J. R. Munkres, *Topology, A First Course*, Prentice-Hall, Englewood Cliffs, New Jersey (1975).

[39] J. R. Munkres, *Elements of Algebraic Topology*, Benjamin/Cummings, Reading, Massachusetts (1984).

[40] H. Okamura and P. Seymour, "Multicommodity flows in planar graphs," *Journal of Combinatorial Theory Series B*, Vol. 31 (1981), pp. 75–81.

[41] R. Y. Pinter, *The Impact of Layer Assignment Methods on Layout Algorithms for Integrated Circuits*, Ph.D. Thesis, MIT Department of Electrical Engineering and Computer Science (August 1982); Technical Report TR-291, MIT Laboratory for Computer Science.

[42] F. P. Preparata and W. Lipski, Jr., "Optimal three-layer channel routing," *IEEE Transactions on Computers*, Vol. C-33 (1984), pp. 350–357.

[43] F. P. Preparata and M. I. Shamos, *Computational Geometry: An Introduction*, Springer-Verlag, New York (1985).

[44] D. Richards, "Complexity of single-layer routing," *IEEE Transactions on Computers*, Vol. C-33, No. 3 (March 1984), pp. 286–288.

[45] R. Rivest, A. Baratz, and G. Miller, "Provably good channel routing algorithms," *Carnegie-Mellon Conference on VLSI Systems and Computations* (October 1981), pp. 153–159.

[46] H. L. Royden, *Real Analysis*, 2nd ed., Macmillan (1968).

[47] W. S. Scott and J. K. Ousterhout, "Plowing: interactive stretching and compaction in Magic," *21st Design Automation Conference* (June 1984), pp. 166–172.

[48] A. Siegel, "River routing: the theory and methodology," Ph.D. thesis, Stanford University (1983).

[49] A. Siegel and D. Dolev, "The separation for general single-layer wiring barriers," *Carnegie-Mellon Conference on VLSI Systems and Computations* (October 1981), pp. 143–152.

[50] E. Spanier, *Algebraic Topology*, 2nd ed., Springer-Verlag, New York (1966).

[51] T. Szymanski, "Dogleg channel routing is NP-complete," *IEEE Transactions on Computer-Aided Design of Circuits and Systems*, Vol. CAD-4, No. 1 (January 1985), pp. 31–41.

[52] M. Tompa, "An optimal solution to a wire-routing problem," *Journal of Computer and System Sciences*, Vol. 23, No. 2 (October 1981), pp. 127–150.

[53] J. Vick, *Homology Theory*, Academic Press (1973).

[54] N. Weste, "MULGA – an interactive symbolic layout system for the design of integrated circuits," *Bell System Technical Journal*, Vol. 60, No. 6 (July-August 1981), pp. 823–857.

[55] T. E. Whitney and C. Mead, "An integer-based hierarchical representation for VLSI," *Fourth MIT Conference on Advanced Research in VLSI* (April 1986), pp. 241–257.

[56] D. E. Willard, "Polygon retrieval," *SIAM Journal on Computing*, Vol. 11 (1982), pp. 149–165.

[57] J. D. Williams, "STICKS – a graphical compiler for high level LSI design," *National Computer Conference* (1978), pp. 289–295.

[58] W. Wolf, "An experimental comparison of 1-D compaction algorithms," *1985 Chapel Hill Conference on VLSI* (May 1985), pp. 165–179.

[59] X.-M. Xiong and E. S. Kuh, "Nutcracker: an efficient and intelligent channel spacer," *24th Design Automation Conference* (1987), pp. 298–304.

OFFICIAL DISTRIBUTION LIST

Director — 2 Copies
Information Processing Techniques Office
Defense Advanced Research Projects Agency
1400 Wilson Boulevard
Arlington, VA 22209

Office of Naval Research — 2 Copies
800 North Quincy Street
Arlington, VA 22217
Attn: Dr. R. Grafton, Code 433

Director, Code 2627 — 6 Copies
Naval Research Laboratory
Washington, DC 20375

Defense Technical Information Center — 12 Copies
Cameron Station
Alexandria, VA 22314

National Science Foundation — 2 Copies
Office of Computing Activities
1800 G. Street, N.W.
Washington, DC 20550
Attn: Program Director

Dr. E.B. Royce, Code 38 — 1 Copy
Head, Research Department
Naval Weapons Center
China Lake, CA 93555

Dr. G. Hooper, USNR — 1 Copy
NAVDAC-OOH
Department of the Navy
Washington, DC 20374

END
Feb.
1988
DTIC